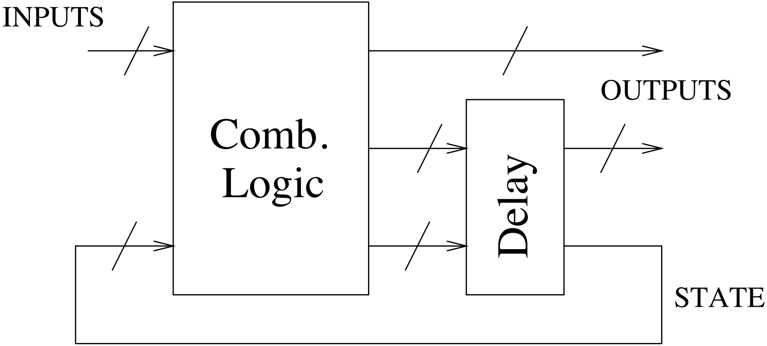
Asynchronous Circuit Design

Chris J. Myers

Lecture 5: Huffman Circuits
Chapter 5

Huffman Circuit



Huffman Circuit Design Method

- *State minimization*
- *State assignment*
- *Logic minimization*

Circuit Delay Model

- Uses the *bounded gate and wire delay model*.
- Environment must also be constrained:
  - *Single-input change* (SIC) - each input change must be separated by a minimum time interval.
  - SIC *fundamental mode* - the time interval is the maximum delay for circuit to stabilize.
  - MIC - allow multiple inputs to change.
  - MIC fundamental mode - waits for circuit to stabilize.
  - Extended burst mode - limited form of MIC operation.

Solving Covering Problems

- The last step of state minimization, state assignment, and logic synthesis is to solve a *covering problem*.
- A covering problem exists whenever you must select a set of choices with minimum cost which satisfy a set of constraints.
- Classic example: selection of the minimum number of prime implicants to cover all the minterms of a given function.

Formal Derivation of Covering Problem

- Each choice is represented with a Boolean variable x_i.
- $x_i = 1$ implies choice has been included in the solution.
- $x_i = 0$ implies choice has not been included in the solution.
- Covering problem is expressed as a product-of-sums, F.
- Each product (or *clause*) represents a constraint.
- Each clause is sum of choices that satisfy the constraint.
- Goal: find x_i's which satisfy all constraints with minimum cost.

$$cost \quad = \quad \min \sum_{i=1}^{t} w_i x_i \tag{1}$$

Example Covering Problem

$$
\begin{aligned}
f \quad = \quad & x_1\overline{x_2}(\overline{x_3}+x_4)(\overline{x_3}+x_4+x_5+x_6)(\overline{x_1}+x_4+x_5+x_6) \\
& (\overline{x_4}+x_1+x_6)(\overline{x_5}+x_6)
\end{aligned}
$$

Unate versus Binate

- *Unate covering problem* - choices appear only in their positive form (i.e., uncomplemented).
- *Binate covering problem* - choices appear in both positive and negative form (i.e., complemented).
- Algorithm presented here considers the more general case of the binate covering problem, but solution applies to both.

Constraint Matrix

- f is represented using a *constraint matrix*, A.
- Includes a column for each x_i variable.
- Includes a row for every clause.
- Each entry of the matrix a_{ij} is:
  - '-' if the variable x_i does not appear in the clause,
  - '0' if the variable appears complemented, and
  - '1' otherwise.
- i^{th} row of A is denoted a_i.
- j^{th} column is denoted by A_j.

Constraint Matrix Example

$$\begin{aligned} f \quad = \quad & x_1\overline{x_2}(\overline{x_3}+x_4)(\overline{x_3}+x_4+x_5+x_6)(\overline{x_1}+x_4+x_5+x_6) \\ & (\overline{x_4}+x_1+x_6)(\overline{x_5}+x_6) \end{aligned}$$

$$\mathbf{A} = \begin{array}{cc} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} & \\ \left[\begin{array}{cccccc} 1 & - & - & - & - & - \\ - & 0 & - & - & - & - \\ - & - & 0 & 1 & - & - \\ - & - & 0 & 1 & 1 & 1 \\ 0 & - & - & 1 & 1 & 1 \\ 1 & - & - & 0 & - & 1 \\ - & - & - & - & 0 & 1 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \end{array}$$

Binate Covering Problem

- The binate covering problem is to find an assignment to **x** of minimum cost such that for every row a_i either
  1. $\exists j \,.\, (a_{ij} = 1) \wedge (x_j = 1)$; or
  2. $\exists j \,.\, (a_{ij} = 0) \wedge (x_j = 0)$.

BCP Algorithm

```
$\mathbf{bcp}(\mathbf{A},\mathbf{x},\mathbf{b})$
  $(\mathbf{A},\mathbf{x})$ = reduce($\mathbf{A},\mathbf{x}$);
  $L =$ lower_bound($\mathbf{A},\mathbf{x}$);
  if ($L \geq$ cost($\mathbf{b}$)) then return($\mathbf{b}$);
  if (terminalCase($\mathbf{A}$)) then
    if ($\mathbf{A}$ has no rows) return($\mathbf{x}$); else return($\mathbf{b}$);
  $c =$ choose_column($\mathbf{A}$);
  $x_c = 1$; $\mathbf{A}^1 =$ select_column($\mathbf{A}, c$); $\mathbf{x}^1 = \mathrm{bcp}(\mathbf{A}^1,\mathbf{x},\mathbf{b})$
  if (cost($\mathbf{x}^1$)<cost($\mathbf{b}$)) then
    $\mathbf{b} = \mathbf{x}^1$;
    if (cost($\mathbf{b}$)$= L$) return($\mathbf{b}$);
  $x_c = 0$; $\mathbf{A}^0 =$ remove_column($\mathbf{A}, c$); $\mathbf{x}^0 = \mathrm{bcp}(\mathbf{A}^0,\mathbf{x},\mathbf{b})$
  if (cost($\mathbf{x}^0$)<cost($\mathbf{b}$)) then $\mathbf{b} = \mathbf{x}^0$;
  return($\mathbf{b}$);
```

Reduce Algorithm

```
reduce(A,x)
  do
    $A' = A$;
    (A,x) = find_essential_rows(A,x);
    A = delete_dominating_rows(A);
    (A,x) = delete_dominated_columns(A,x);
  while (A $\neq \emptyset$ and A $\neq$ A');
  return(A,x);
```

Essential Rows

- A row a_i of A is *essential* when there exists exactly one j such that a_{ij} is not equal to '-'.
- This cooresponds to clause consisting of a single literal.
- If the literal is x_j (i.e., $a_{ij} = 1$), the variable is *essential*.
- If the literal is $\overline{x_j}$ (i.e., $a_{ij} = 0$), the variable is *unacceptable*.
- The matrix A is reduced with respect to the essential literal.
- This variable is set to value of literal, column is removed, and any row where variable has same value is removed.

Essential Rows Example

$$\begin{aligned} f \quad = \quad & x_1\overline{x_2}(\overline{x_3}+x_4)(\overline{x_3}+x_4+x_5+x_6)(\overline{x_1}+x_4+x_5+x_6) \\ & (\overline{x_4}+x_1+x_6)(\overline{x_5}+x_6) \end{aligned}$$

$$\mathbf{A} = \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \left[\begin{array}{cccccc} 1 & - & - & - & - & - \\ - & 0 & - & - & - & - \\ - & - & 0 & 1 & - & - \\ - & - & 0 & 1 & 1 & 1 \\ 0 & - & - & 1 & 1 & 1 \\ 1 & - & - & 0 & - & 1 \\ - & - & - & - & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

Essential Rows Example

$$f = \overline{x_2}(\overline{x_3} + x_4)(\overline{x_3} + x_4 + x_5 + x_6)(x_4 + x_5 + x_6)(\overline{x_5} + x_6)$$

$$\mathbf{A} = \begin{array}{c c} \begin{array}{ccccc} x_2 & x_3 & x_4 & x_5 & x_6 \end{array} & \\ \left[\begin{array}{ccccc} 0 & - & - & - & - \\ - & 0 & 1 & - & - \\ - & 0 & 1 & 1 & 1 \\ - & - & 1 & 1 & 1 \\ - & - & - & 0 & 1 \end{array}\right] & \begin{array}{c} 2 \\ 3 \\ 4 \\ 5 \\ 7 \end{array} \end{array}$$

$$x_1 = 1$$

Essential Rows Example

$$f = (\overline{x_3} + x_4)(\overline{x_3} + x_4 + x_5 + x_6)(x_4 + x_5 + x_6)(\overline{x_5} + x_6)$$

$$\mathbf{A} = \begin{array}{cc} \begin{array}{cccc} x_3 & x_4 & x_5 & x_6 \end{array} & \\ \left[\begin{array}{cccc} 0 & 1 & - & - \\ 0 & 1 & 1 & 1 \\ - & 1 & 1 & 1 \\ - & - & 0 & 1 \end{array}\right] & \begin{array}{c} 3 \\ 4 \\ 5 \\ 7 \end{array} \end{array}$$

$$x_1 = 1, x_2 = 0$$

Row Dominance

- A row a_k *dominates* another row a_i if it has all 1's and 0's of a_i.
- Row a_k dominates another row a_i if for each column A_j of **A**, one of the following is true:
  - $a_{ij} = -$
  - $a_{ij} = a_{kj}$
- Removing dominating rows does not affect set of solutions.

Row Dominance Example

$$f = (\overline{x_3} + x_4)(\overline{x_3} + x_4 + x_5 + x_6)(x_4 + x_5 + x_6)(\overline{x_5} + x_6)$$

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{cccc} x_3 & x_4 & x_5 & x_6 \\ \end{array}$$

$$\mathbf{A} = \begin{array}{cccccc} & x_3 & x_4 & x_5 & x_6 & \\ \left[\right. & 0 & 1 & - & - & \left.\right] \quad 3 \\ & 0 & 1 & 1 & 1 & \quad 4 \\ & - & 1 & 1 & 1 & \quad 5 \\ & - & - & 0 & 1 & \quad 7 \end{array}$$

$$x_1 = 1, x_2 = 0$$

Row Dominance Example

$$f = (\overline{x_3} + x_4)(x_4 + x_5 + x_6)(\overline{x_5} + x_6)$$

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{cccc} x_3 & x_4 & x_5 & x_6 \\ \end{array}$$

$$\mathbf{A} = \left[\begin{array}{cccc} 0 & 1 & - & - \\ - & 1 & 1 & 1 \\ - & - & 0 & 1 \end{array}\right] \begin{array}{c} 3 \\ 5 \\ 7 \end{array}$$

$$x_1 = 1, x_2 = 0$$

Column Dominance

- A column A_j *dominates* another column A_k if for each clause a_i of A, one of the following is true:
    - $a_{ij} = 1$;
    - $a_{ij} = -$ and $a_{ik} \neq 1$;
    - $a_{ij} = 0$ and $a_{ik} = 0$.
- Dominated columns can be removed without affecting the existence of a solution.
- When removing a column, the variable is set to 0 which means any rows including that column with a 0 entry can be removed.

Column Dominance Example

$$f = (\overline{x_3} + x_4)(x_4 + x_5 + x_6)(\overline{x_5} + x_6)$$

$$\mathbf{A} = \begin{array}{cccc|c} x_3 & x_4 & x_5 & x_6 & \\ 0 & 1 & - & - & 3 \\ - & 1 & 1 & 1 & 5 \\ - & - & 0 & 1 & 7 \end{array}$$

$$x_1 = 1, x_2 = 0$$

Column Dominance Example

$$f = (x_4 + x_6)$$

$$\mathbf{A} = \begin{array}{cc} x_4 & x_6 \\ \left[\begin{array}{c} 1 \end{array}\right. & \left.\begin{array}{c} 1 \end{array}\right] \end{array} \quad 5$$

$$x_1 = 1, x_2 = 0, x_3 = 0, x_5 = 0$$

Checking Weights

- If weights are not equal, it is necessary to also check the weights of the columns before removing dominated columns.
- If weight of dominating column, w_j, is greater than weight of dominated column, w_k, then x_k should not be removed.
- Assume $w_1 = 3$, $w_2 = 1$, and $w_3 = 1$.

$$\mathbf{A} = \begin{array}{cc} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} & \\ \left[\begin{array}{ccc} 1 & 1 & - \\ - & 0 & 1 \end{array}\right] & \begin{array}{c} 1 \\ 2 \end{array} \end{array}$$

BCP Algorithm

```
bcp(A,x,b)
  (A,x) = reduce(A,x);
  $L$ = lower_bound(A,x);
  if ($L \geq$ cost(b)) then return(b);
  if (terminalCase(A)) then
    if (A has no rows) return(x); else return(b);
  $c$ = choose_column(A);
  $x_c = 1$; $A^1$ = select_column(A,c); $x^1$ = bcp($A^1$,x,b)
  if (cost($x^1$)<cost(b)) then
    b = $x^1$;
    if (cost(b)=$L$) return(b);
  $x_c = 0$; $A^0$ = remove_column(A,c); $x^0$ = bcp($A^0$,x,b)
  if (cost($x^0$)<cost(b)) then b = $x^0$;
  return(b);
```

Bounding

- If solved, cost of solution can be determined by Equation 1.
- Reduced matrix may have a *cyclic core*.
- Must test whether or not a good solution can be derived from partial solution found up to this point.
- Determine a lower bound, L, on the final cost, starting with the current partial solution.
- If L is greater than or equal to the cost of the best solution found, the previous best solution is returned.

Maximal Independent Set

- Finding exact lower bound is as difficult as solving the covering problem.
- Satisfactory heuristic method is to find a *maximal independent set* (MIS) of rows.
- Two rows are independent when it is not possible to satisfy both by setting a single variable to 1.
- Any row which contains a complemented variable is dependent on any other clause, so we must ignore these rows.

Lower Bound Algorithm

```
lower_bound(A, x)
  MIS = $\emptyset$
  A = delete_rows_with_complemented_variables(A);
  do
    i =choose_shortest_row(A);
    MIS = MIS $\cup$ {i};
    A =delete_intersecting_rows(A, i);
  while (A $\neq \emptyset$);
  return(|MIS| + cost(x));
```

Bounding Example

$$\mathbf{A} = \begin{array}{cc}
\begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} & \\
\left[\begin{array}{ccccccccc}
1 & 1 & - & - & - & - & - & - & - \\
1 & - & 1 & - & - & - & - & - & - \\
- & - & - & 1 & 1 & - & - & - & - \\
- & - & - & 1 & - & 1 & - & - & - \\
- & - & 1 & - & 1 & 1 & - & - & - \\
- & - & 1 & - & - & - & 1 & - & - \\
- & 1 & - & - & - & - & 1 & - & - \\
- & - & - & 1 & - & - & - & 1 & - \\
- & - & - & 1 & - & - & - & - & 1 \\
- & 1 & - & - & - & - & - & 1 & 1
\end{array}\right] &
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\end{array}$$

Bounding Example

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{ccccccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & \\ - & - & - & 1 & 1 & - & - & - & - & 3 \\ - & - & - & 1 & - & 1 & - & - & - & 4 \\ - & - & 1 & - & 1 & 1 & - & - & - & 5 \\ - & - & 1 & - & - & - & 1 & - & - & 6 \\ - & - & - & 1 & - & - & - & 1 & - & 8 \\ - & - & - & 1 & - & - & - & - & 1 & 9 \end{array}$$

$$MIS = \{1\}$$

Bounding Example

$$\mathbf{A} = \begin{array}{c} \\ \end{array} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \\ \left[- \right. & - & 1 & - & - & - & 1 & - & \left. - \right] \end{array} \quad 6$$

$$MIS = \{1, 3\}$$

Bounding Example

$$\mathbf{A} = \begin{array}{cc}
\begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} & \\
\left[\begin{array}{ccccccccc}
1 & 1 & - & - & - & - & - & - & - \\
1 & - & 1 & - & - & - & - & - & - \\
- & - & - & 1 & 1 & - & - & - & - \\
- & - & - & 1 & - & 1 & - & - & - \\
- & - & 1 & - & 1 & 1 & - & - & - \\
- & - & 1 & - & - & - & 1 & - & - \\
- & 1 & - & - & - & - & 1 & - & - \\
- & - & - & 1 & - & - & - & 1 & - \\
- & - & - & 1 & - & - & - & - & 1 \\
- & 1 & - & - & - & - & - & 1 & 1
\end{array}\right] &
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\end{array}$$

$$MIS = \{1, 3, 6\}$$

BCP Algorithm

```
bcp($\mathbf{A},\mathbf{x},\mathbf{b}$)
  ($\mathbf{A},\mathbf{x}$) = reduce($\mathbf{A},\mathbf{x}$);
  $L =$ lower_bound($\mathbf{A},\mathbf{x}$);
  if ($L \geq$ cost($\mathbf{b}$)) then return($\mathbf{b}$);
  if (terminalCase($\mathbf{A}$)) then
    if ($\mathbf{A}$ has no rows) return($\mathbf{x}$); else return($\mathbf{b}$);
  $c =$ choose_column($\mathbf{A}$);
  $x_c = 1$; $\mathbf{A^1} =$ select_column($\mathbf{A}, c$); $\mathbf{x^1} =$ bcp($\mathbf{A^1},\mathbf{x},\mathbf{b}$)
  if (cost($\mathbf{x^1}$)<cost($\mathbf{b}$)) then
    $\mathbf{b} = \mathbf{x^1}$;
    if (cost($\mathbf{b}$)$= L$) return($\mathbf{b}$);
  $x_c = 0$; $\mathbf{A^0} =$ remove_column($\mathbf{A}, c$); $\mathbf{x^0} =$ bcp($\mathbf{A^0},\mathbf{x},\mathbf{b}$)
  if (cost($\mathbf{x^0}$)<cost($\mathbf{b}$)) then $\mathbf{b} = \mathbf{x^0}$;
  return($\mathbf{b}$);
```

Termination

- If A has no more rows, then all the constraints have been satisfied by x, and it is a terminal case.
- If no solution exists, it is also a terminal case.

Infeasible Problems

$$f = (x_1 + x_2)(\overline{x_1} + x_2)(x_1 + \overline{x_2})(\overline{x_1} + \overline{x_2})$$

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{cc} x_1 & x_2 \\ \end{array}$$

$$\mathbf{A} = \begin{array}{cc|c} x_1 & x_2 & \\ \hline 1 & 1 & 1 \\ 0 & 1 & 2 \\ 1 & 0 & 3 \\ 0 & 0 & 4 \end{array}$$

BCP Algorithm

```
bcp(A, x, b)
  (A, x) = reduce(A, x);
  $L = $ lower_bound(A, x);
  if ($L \geq$ cost(b)) then return(b);
  if (terminalCase(A)) then
    if (A has no rows) return(x); else return(b);
  $c = $ choose_column(A);
  $x_c = 1$; $A^1 = $ select_column(A, $c$); $x^1 = $ bcp($A^1$, x, b)
  if (cost($x^1$) < cost(b)) then
    $b = x^1$;
    if (cost(b) = $L$) return(b);
  $x_c = 0$; $A^0 = $ remove_column(A, $c$); $x^0 = $ bcp($A^0$, x, b)
  if (cost($x^0$) < cost(b)) then $b = x^0$;
  return(b);
```

Branching

- If **A** is not a terminal case, matrix is *cyclic*.
- To find minimal solution, must determine column to branch on.
- A column intersecting short rows is prefered for branching.
- Assign a weight to each row that is inverse of row length.
- Sum the weights of all the rows covered by a column.
- Column x_c with highest value is chosen for case splitting.

Branching Example

$$\mathbf{A} = \begin{array}{cc} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} & \\ \left[\begin{array}{ccccccccc} 1 & 1 & - & - & - & - & - & - & - \\ 1 & - & 1 & - & - & - & - & - & - \\ - & - & - & 1 & 1 & - & - & - & - \\ - & - & - & 1 & - & 1 & - & - & - \\ - & - & 1 & - & 1 & 1 & - & - & - \\ - & - & 1 & - & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 & - & - \\ - & - & - & 1 & - & - & - & 1 & - \\ - & - & - & 1 & - & - & - & - & 1 \\ - & 1 & - & - & - & - & - & 1 & 1 \end{array}\right] & \begin{array}{r} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \end{array}$$

Branching Example

$$\mathbf{A} = \begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} \\ \left[\begin{array}{ccccccccc} 1 & 1 & - & - & - & - & - & - & - \\ 1 & - & 1 & - & - & - & - & - & - \\ - & - & - & 1 & 1 & - & - & - & - \\ - & - & - & 1 & - & 1 & - & - & - \\ - & - & 1 & - & 1 & 1 & - & - & - \\ - & - & 1 & - & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 & - & - \\ - & - & - & 1 & - & - & - & 1 & - \\ - & - & - & 1 & - & - & - & - & 1 \\ - & 1 & - & - & - & - & - & 1 & 1 \end{array}\right] \end{array} \quad \begin{array}{cc} \\ 1 & 1/2 \\ 2 & 1/2 \\ 3 & 1/2 \\ 4 & 1/2 \\ 5 & 1/3 \\ 6 & 1/2 \\ 7 & 1/2 \\ 8 & 1/2 \\ 9 & 1/2 \\ 10 & 1/3 \end{array}$$

Branching Example

$$\mathbf{A} = \begin{array}{ccccccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & & \\
1.0 & 1.3 & 1.3 & 2.0 & 0.8 & 0.8 & 1.0 & 0.8 & 0.8 & & \\
1 & 1 & - & - & - & - & - & - & - & 1 & 1/2 \\
1 & - & 1 & - & - & - & - & - & - & 2 & 1/2 \\
- & - & - & 1 & 1 & - & - & - & - & 3 & 1/2 \\
- & - & - & 1 & - & 1 & - & - & - & 4 & 1/2 \\
- & - & 1 & - & 1 & 1 & - & - & - & 5 & 1/3 \\
- & - & 1 & - & - & - & 1 & - & - & 6 & 1/2 \\
- & 1 & - & - & - & - & 1 & - & - & 7 & 1/2 \\
- & - & - & 1 & - & - & - & 1 & - & 8 & 1/2 \\
- & - & - & 1 & - & - & - & - & 1 & 9 & 1/2 \\
- & 1 & - & - & - & - & - & 1 & 1 & 10 & 1/3
\end{array}$$

Branching

- x_c is added to the solution and constraint matrix is reduced.
- *bcp* is called recursively and result assigned to $\mathbf{x}^1$.
- If $\mathbf{x}^1$ better than best, record it.
- If $\mathbf{x}^1$ meets lower bound L, it is minimal.
- If not, remove x_c from solution and call *bcp*.
- If $\mathbf{x}^0$ better than best, return it.

BCP Algorithm

```
$\mathbf{bcp}(\mathbf{A},\mathbf{x},\mathbf{b})$
  $(\mathbf{A},\mathbf{x})$ = reduce$(\mathbf{A},\mathbf{x})$;
  $L =$ lower_bound$(\mathbf{A},\mathbf{x})$;
  if ($L \geq$ cost($\mathbf{b}$)) then return($\mathbf{b}$);
  if (terminalCase($\mathbf{A}$)) then
    if ($\mathbf{A}$ has no rows) return($\mathbf{x}$); else return($\mathbf{b}$);
  $c =$ choose_column($\mathbf{A}$);
  $x_c = 1$; $\mathbf{A}^1 =$ select_column($\mathbf{A}, c$); $\mathbf{x}^1 =$ bcp$(\mathbf{A}^1,\mathbf{x},\mathbf{b})$
  if (cost($\mathbf{x}^1$)<cost($\mathbf{b}$)) then
    $\mathbf{b} = \mathbf{x}^1$;
    if (cost($\mathbf{b}$)$= L$) return($\mathbf{b}$);
  $x_c = 0$; $\mathbf{A}^0 =$ remove_column($\mathbf{A}, c$); $\mathbf{x}^0 =$ bcp$(\mathbf{A}^0,\mathbf{x},\mathbf{b})$
  if (cost($\mathbf{x}^0$)<cost($\mathbf{b}$)) then $\mathbf{b} = \mathbf{x}^0$;
  return($\mathbf{b}$);
```

Branching Example

$$\mathbf{A} = \begin{array}{cccccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & \\
1 & 1 & - & - & - & - & - & - & - & 1 \\
1 & - & 1 & - & - & - & - & - & - & 2 \\
- & - & - & 1 & 1 & - & - & - & - & 3 \\
- & - & - & 1 & - & 1 & - & - & - & 4 \\
- & - & 1 & - & 1 & 1 & - & - & - & 5 \\
- & - & 1 & - & - & - & 1 & - & - & 6 \\
- & 1 & - & - & - & - & 1 & - & - & 7 \\
- & - & - & 1 & - & - & - & 1 & - & 8 \\
- & - & - & 1 & - & - & - & - & 1 & 9 \\
- & 1 & - & - & - & - & - & 1 & 1 & 10
\end{array}$$

Branching Example

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccccc|c} x_1 & x_2 & x_3 & x_5 & x_6 & x_7 & x_8 & x_9 & \\ 1 & 1 & - & - & - & - & - & - & 1 \\ 1 & - & 1 & - & - & - & - & - & 2 \\ - & - & 1 & 1 & 1 & - & - & - & 5 \\ - & - & 1 & - & - & 1 & - & - & 6 \\ - & 1 & - & - & - & 1 & - & - & 7 \\ - & 1 & - & - & - & - & 1 & 1 & 10 \end{array}$$

$$x_4 = 1$$

Branching Example

$$\mathbf{A} = \begin{array}{c c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_7 \end{array} & \\ \left[\begin{array}{cccc} 1 & 1 & - & - \\ 1 & - & 1 & - \\ - & - & 1 & - \\ - & - & 1 & 1 \\ - & 1 & - & 1 \\ - & 1 & - & - \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 5 \\ 6 \\ 7 \\ 10 \end{array} \end{array}$$

$$x_4 = 1, x_5 = 0, x_6 = 0, x_8 = 0, x_9 = 0$$

Branching Example

$$x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 0, x_6 = 0, x_8 = 0, x_9 = 0$$

cost($\mathbf{x}^1$) = 3

Recall that $L = 3$

Therefore, we are done.

Branching Example

$$\mathbf{A} = \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & \\ 1 & 1 & - & - & - & - & - & - & - & 1 \\ 1 & - & 1 & - & - & - & - & - & - & 2 \\ - & - & - & 1 & 1 & - & - & - & - & 3 \\ - & - & - & 1 & - & 1 & - & - & - & 4 \\ - & - & 1 & - & 1 & 1 & - & - & - & 5 \\ - & - & 1 & - & - & - & 1 & - & - & 6 \\ - & 1 & - & - & - & - & 1 & - & - & 7 \\ - & - & - & 1 & - & - & - & 1 & - & 8 \\ - & - & - & 1 & - & - & - & - & 1 & 9 \\ - & 1 & - & - & - & - & - & 1 & 1 & 10 \end{array}$$

Branching Example

$$\mathbf{A} = \begin{array}{c} \\ \\ \end{array} \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_5 & x_6 & x_7 & x_8 & x_9 \\ \end{array}$$

$$\mathbf{A} = \begin{array}{cc} \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} & \\ \left[\begin{array}{cccccccc} 1 & 1 & - & - & - & - & - & - \\ 1 & - & 1 & - & - & - & - & - \\ - & - & - & 1 & - & - & - & - \\ - & - & - & - & 1 & - & - & - \\ - & - & 1 & 1 & 1 & - & - & - \\ - & - & 1 & - & - & 1 & - & - \\ - & 1 & - & - & - & 1 & - & - \\ - & - & - & - & - & - & 1 & - \\ - & - & - & - & - & - & - & 1 \\ - & 1 & - & - & - & - & 1 & 1 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \end{array}$$

$$x_4 = 0$$

Branching Example

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{cccc} x_1 & x_2 & x_3 & x_7 \\ \hline 1 & 1 & - & - \\ 1 & - & 1 & - \\ - & - & 1 & 1 \\ - & 1 & - & 1 \end{array} \quad \begin{array}{c} \\ 1 \\ 2 \\ 6 \\ 7 \end{array}$$

$$x_4 = 0, x_5 = 1, x_6 = 1, x_8 = 1, x_9 = 1$$

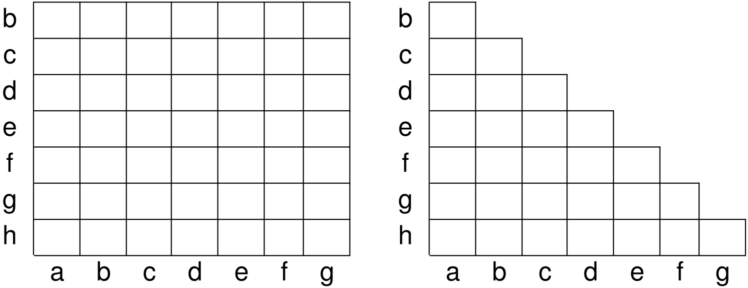
State Minimization Overview

- Original flow table may contain redundant rows, or states.
- Reducing number of states, reduces number of state variables.
- State minimization procedure:
    - Identify all *compatible pairs* of states.
    - Finds all *maximal compatibles*.
    - Find set of *prime compatibles*.
    - Setup a covering problem where prime compatibles are the solutions, and states are what needs to be covered.
- For SIC fundamental mode, same as for synchronous FSMs.

Example Huffman Flow Table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
a	a,0	–	d,0	e,1	b,0	a,–	–
b	b,0	d,1	a,–	–	a,–	a,1	–
c	b,0	d,1	a,1	–	–	–	g,0
d	–	e,–	–	b,–	b,0	–	a,–
e	b,–	e,–	a,–	–	b,–	e,–	a,1
f	b,0	c,–	–,1	h,1	f,1	g,0	–
g	–	c,1	–	e,1	–	g,0	f,0
h	a,1	e,0	d,1	b,0	b,–	e,–	a,1

Pair Chart



Unconditionally Compatible

- Two states u and v are *output compatible* when for each input in which both are specified, they produce the same output.
- Two states u and v are *unconditionally compatible* when output compatible and go to the same next states.
- When two states u and v are unconditionally compatible, the (u,v) entry is marked with the symbol $\sim$.

Example Huffman Flow Table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
a	a,0	–	d,0	e,1	b,0	a,–	–
b	b,0	d,1	a,–	–	a,–	a,1	–
c	b,0	d,1	a,1	–	–	–	g,0
d	–	e,–	–	b,–	b,0	–	a,–
e	b,–	e,–	a,–	–	b,–	e,–	a,1
f	b,0	c,–	–,1	h,1	f,1	g,0	–
g	–	c,1	–	e,1	–	g,0	f,0
h	a,1	e,0	d,1	b,0	b,–	e,–	a,1

Example after Marking Unconditional Compatibles

b							
c		∼					
d							
e				∼			
f							
g	∼						
h				∼			
	a	b	c	d	e	f	g

Incompatibles

- When two states *u* and *v* are not output compatible, the states are *incompatible*.
- When two states *u* and *v* are incompatible, the (*u*,*v*) entry is marked with the symbol ×.

Example Huffman Flow Table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
a	a,0	–	d,0	e,1	b,0	a,–	–
b	b,0	d,1	a,–	–	a,–	a,1	–
c	b,0	d,1	a,1	–	–	–	g,0
d	–	e,–	–	b,–	b,0	–	a,–
e	b,–	e,–	a,–	–	b,–	e,–	a,1
f	b,0	c,–	–,1	h,1	f,1	g,0	–
g	–	c,1	–	e,1	–	g,0	f,0
h	a,1	e,0	d,1	b,0	b,–	e,–	a,1

Example after Marking Incompatibles

	a	b	c	d	e	f	g
b							
c	×	∼					
d							
e			×	∼			
f	×	×		×			
g	∼	×			×		
h	×	×	×	∼		×	×

Conditionally Compatible

- Two states are *conditionally compatible* when there exists differences in their next state entries.
- If differing next states are merged, they become compatible.
- When two states u and v are compatible only when states s and t are merged then the (u,v) entry is marked with s,t.

Example Huffman Flow Table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
a	a,0	–	d,0	e,1	b,0	a,–	–
b	b,0	d,1	a,–	–	a,–	a,1	–
c	b,0	d,1	a,1	–	–	–	g,0
d	–	e,–	–	b,–	b,0	–	a,–
e	b,–	e,–	a,–	–	b,–	e,–	a,1
f	b,0	c,–	–,1	h,1	f,1	g,0	–
g	–	c,1	–	e,1	–	g,0	f,0
h	a,1	e,0	d,1	b,0	b,–	e,–	a,1

Example after Marking Conditional Compatibles

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	c,e b,f e,g		
g	∼	×	c,d f,g	c,e b,e a,f	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

Final Check

- The final step is to check each pair of conditional compatibles.
- If any pair of next states are known to be incompatible, then the states are are also incompatible.
- In this case, the (u,v) entry is marked with the symbol $\times$.

Example after Marking Conditional Compatibles

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	c,e b,f e,g		
g	∼	×	c,d f,g	c,e b,e a,f	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

Final Pair Chart

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

Maximal Compatibles

- Next need to find larger sets of compatible states.
- If S is compatible, then any subset of S is also compatible.
- A *maximal compatible* is a compatible that is not a subset of any larger compatible.
- From maximal compatibles, can determine all other compatibles.

Approach One

- Initialize compatible list (*c*-list) with compatible pairs in rightmost column of pair chart having at least one non-× entry.
- Examine the columns from right to left.
- Set S_i to states in column i which do not contain ×.
- Intersect S_i with each member of the current *c*-list.
- If the intersection has more than one member, add to the *c*-list an entry composed of the intersection unioned with i.
- Remove duplicate entries and those that are subset of others.
- Add pairs which consist of i and any members of S_i that did not appear in any of the intersections.
- *c*-list plus states not contained in *c*-list are maximal compatibles.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

Initialize compatible list (*c*-list) with compatible pairs in rightmost column of pair chart having at least one non-× entry.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

Initialize compatible list (c-list) with compatible pairs in rightmost column of pair chart having at least one non-× entry.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

Examine the columns from right to left.

Example for Approach One

b	a,d						
c	×	$\sim$					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	$\sim$			
f	×	×	c,d	×	×		
g	$\sim$	×	c,d f,g	×	×	e,h	
h	×	×	×	$\sim$	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e =$

Set S_i to states in column i which do not contain ×.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$

Set S_i to states in column i which do not contain ×.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$

Intersect S_i with each member of the current c-list, add to the c-list an entry composed of the intersection unioned with i.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$

Add pairs which consist of i and any members of S_i that did not appear in any of the intersections.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

Add pairs which consist of *i* and any members of S_i that did not appear in any of the intersections.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d =$

Set S_i to states in column i which do not contain ×.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$

Set S_i to states in column i which do not contain $\times$.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h :$ $c = \{fg, eh\}$

$S_d = eh :$

Intersect S_i with each member of the current c-list, add to the c-list an entry composed of the intersection unioned with i.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h :$ $c = \{fg, eh\}$

$S_d = eh :$ $c = \{fg, eh, deh\}$

Example for Approach One

b	a,d						
c	×	$\sim$					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	$\sim$			
f	×	×	c,d	×	×		
g	$\sim$	×	c,d f,g	×	×	e,h	
h	×	×	×	$\sim$	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$ $c = \{fg, eh, deh\}$

Remove duplicate entries and those that are subset of others.

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$ $c = \{fg, deh\}$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$ $c = \{fg, deh\}$

$S_c =$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h$: $c = \{fg, eh\}$

$S_d = eh$: $c = \{fg, deh\}$

$S_c = dfg$:

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$ $c = \{fg, deh\}$

$S_c = dfg:$ $c = \{cfg, deh, cd\}$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$ $c = \{fg, deh\}$

$S_c = dfg:$ $c = \{cfg, deh, cd\}$

$S_b =$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$ $c = \{fg, deh\}$

$S_c = dfg:$ $c = \{cfg, deh, cd\}$

$S_b = cde:$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h :$ $c = \{fg, eh\}$

$S_d = eh :$ $c = \{fg, deh\}$

$S_c = dfg :$ $c = \{cfg, deh, cd\}$

$S_b = cde :$ $c = \{cfg, deh, bcd, bde\}$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h:$ $c = \{fg, eh\}$

$S_d = eh:$ $c = \{fg, deh\}$

$S_c = dfg:$ $c = \{cfg, deh, cd\}$

$S_b = cde:$ $c = \{cfg, deh, bcd, bde\}$

$S_a =$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h :$ $c = \{fg, eh\}$

$S_d = eh :$ $c = \{fg, deh\}$

$S_c = dfg :$ $c = \{cfg, deh, cd\}$

$S_b = cde :$ $c = \{cfg, deh, bcd, bde\}$

$S_a = bdeg :$ $c = \{cfg, deh, bcd, abde, ag\}$

Example for Approach One

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

First step: $c = \{fg\}$

$S_e = h$: $c = \{fg, eh\}$

$S_d = eh$: $c = \{fg, deh\}$

$S_c = dfg$: $c = \{cfg, deh, cd\}$

$S_b = cde$: $c = \{cfg, deh, bcd, bde\}$

$S_a = bdeg$: $c = \{cfg, deh, bcd, abde, ag\}$

c-list plus states not contained in *c*-list are maximal compatibles.

Approach Two

- If s_i and s_j have been found to be incompatible, we know that no maximal compatible can include both.
- Write a Boolean formula that gives the conditions for a set of states to be compatible.
- For each state s_i, $x_i = 1$ means that s_i is in the set.
- States s_i, s_j incompatible implies clause $(\overline{x_i} + \overline{x_j})$ is included.
- Form conjunction of clauses for each incompatible pair.

Example for Approach Two

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

Example for Approach Two

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

$$(\overline{a}+\overline{c})(\overline{a}+\overline{f})(\overline{a}+\overline{h})(\overline{b}+\overline{f})(\overline{b}+\overline{g})(\overline{b}+\overline{h})(\overline{c}+\overline{e})$$

$$(\overline{c}+\overline{h})(\overline{d}+\overline{f})(\overline{d}+\overline{g})(\overline{e}+\overline{f})(\overline{e}+\overline{g})(\overline{f}+\overline{h})(\overline{g}+\overline{h})$$

Example for Approach Two

- Initial Boolean formula for incompatibles:

$$(\overline{a}+\overline{c})(\overline{a}+\overline{f})(\overline{a}+\overline{h})(\overline{b}+\overline{f})(\overline{b}+\overline{g})(\overline{b}+\overline{h})(\overline{c}+\overline{e})$$
$$(\overline{c}+\overline{h})(\overline{d}+\overline{f})(\overline{d}+\overline{g})(\overline{e}+\overline{f})(\overline{e}+\overline{g})(\overline{f}+\overline{h})(\overline{g}+\overline{h})$$

- Convert to sum-of-products:

$$\overline{a}\,\overline{b}\,\overline{d}\,\overline{e}\,\overline{h}+\overline{a}\,\overline{b}\,\overline{c}\,\overline{f}\,\overline{g}+\overline{a}\,\overline{e}\,\overline{f}\,\overline{g}\,\overline{h}+\overline{c}\,\overline{f}\,\overline{g}\,\overline{h}+\overline{b}\,\overline{c}\,\overline{d}\,\overline{e}\,\overline{f}\,\overline{h}$$

- Each term defines a maximal compatible set where states that do not occur make up the maximal compatible.

$$cfg, deh, bcd, abde, ag$$

Prime Compatibles

- Some states are compatible only if other pairs are merged.
- The implied state set for each compatible is called its *class set*.
- The implied compatibles must be selected to guarantee *closure*.
- C_1 and C_2 are compatibles and Γ_1 and Γ_2 are their class sets.
- If $C_1 \subset C_2$ then it may appear that C_2 is better, but if $\Gamma_1 \subset \Gamma_2$ then C_1 may be better.
- The best compatibles may not be maximal.
- A compatible C_1 is *prime* iff there does not exist $C_2 \supset C_1$ such that $\Gamma_2 \subseteq \Gamma_1$.
- An optimum solution always uses only prime compatibles.

Prime Compatible Algorithm

```
prime_compatibles($C, M$) $done = \emptyset$
    for($k = |largest(M)|;\ k \geq 1;\ k--$)
        foreach($q \in M;\ |q| = k$) $enqueue(P, q)$
        foreach($p \in P;\ |p| = k$)
            if($class\_set(CM, p) = \emptyset$) continue
            foreach($s \in max\_subsets(p)$)
                if($s \in done$) continue
                $\Gamma_s = class\_set(CM, s)$
                $prime = true$
                foreach($q \in P; |q| \geq k$)
                    if ($s \subset q$)
                        $\Gamma_q = class\_set(CM, q)$
                        if ($\Gamma_s \supseteq \Gamma_q$)
                            $prime = false$; break
                if($prime = 1$) enqueue($P, s$)
                $done = done \cup \{s\}$
```

Example for Prime Compatibles

b	a,d						
c	×	∼					
d	b,e	a,b d,e	d,e a,g				
e	a,b a,d	d,e a,b a,e	×	∼			
f	×	×	c,d	×	×		
g	∼	×	c,d f,g	×	×	e,h	
h	×	×	×	∼	a,b a,d	×	×
	a	b	c	d	e	f	g

Maximal compatiables $= \{abde, bcd, cfg, deh, ag\}$

Prime Compatibles

	Prime compatibles	Class set
1	*abde*	$\emptyset$
2	*bcd*	$\{(a,b),(a,g),(d,e)\}$
3	*cfg*	$\{(c,d),(e,h)\}$
4	*deh*	$\{(a,b),(a,d)\}$
5	*bc*	$\emptyset$
6	*cd*	$\{(a,g),(d,e)\}$
7	*cf*	$\{(c,d)\}$
8	*cg*	$\{(c,d),(f,g)\}$
9	*fg*	$\{(e,h)\}$
10	*dh*	$\emptyset$
11	*ag*	$\emptyset$
12	*f*	$\emptyset$

Setting up the Covering Problem

- A collection of prime compatibles forms a valid solution when it is a *closed cover*.
- A collection of compatibles is a *cover* when all states are contained in some compatible in the set.
- A collection is *closed* when all implied states are contained in some other compatible.
- $c_i = 1$ when the i^{th} prime compatible is in the solution.
- Using c_i variables, can write a Boolean formula that represents the conditions for a solution to be a closed cover.
- The formula is a product-of-sums where each product is a covering or closure constraint.

Covering Constraints

- There is one covering constraint for each state.
- The product is simply a disjunction of the prime compatibles that include the state.
- In other words, for the covering constraint to yield 1, one of the primes that includes the state must be in the solution. For example, the covering constraint for state *a* is:

$$(c_1 + c_{11})$$

Closure Constraints

- There is a closure constraint for each implied compatible for each prime compatible.
- For example, the prime *bcd* requires the following states to be merged: (a,b), (a,g), (d,e).
- Therefore, if we include *bcd* in the cover (i.e., c_2), then we must also select compatibles which will merge these other state pairs.
- *abde* is the only prime compatible that merges a and b.
- Therefore, we have a closure constraint of the form:

$$c_2 \Rightarrow c_1$$

Closure Constraints

- The prime *ag* is the only one that merges states *a* and *g*, so we also need a closure constraint of the form:

$$c_2 \Rightarrow c_{11}$$

- Finally, primes *abde* and *deh* both merge states *d* and *e*, so the resulting closure constraint is:

$$c_2 \Rightarrow (c_1 + c_4)$$

- Converting the implication into disjunctions, we can express the complete set of closure constraints for *bcd* as follows:

$$(\overline{c_2} + c_1)(\overline{c_2} + c_{11})(\overline{c_2} + c_1 + c_4)$$

Prime Compatibles

	Prime compatibles	Class set
1	$abde$	$\emptyset$
2	bcd	$\{(a,b),(a,g),(d,e)\}$
3	cfg	$\{(c,d),(e,h)\}$
4	deh	$\{(a,b),(a,d)\}$
5	bc	$\emptyset$
6	cd	$\{(a,g),(d,e)\}$
7	cf	$\{(c,d)\}$
8	cg	$\{(c,d),(f,g)\}$
9	fg	$\{(e,h)\}$
10	dh	$\emptyset$
11	ag	$\emptyset$
12	f	$\emptyset$

Product-of-Sums Formulation

$$
\begin{aligned}
& (c_1 + c_{11})(c_1 + c_2 + c_5)(c_2 + c_3 + c_5 + c_6 + c_7 + c_8) \\
& \cdot(c_1 + c_2 + c_4 + c_6 + c_{10})(c_1 + c_4)(c_3 + c_7 + c_9 + c_{12}) \\
& \cdot(c_3 + c_8 + c_9 + c_{11})(c_4 + c_{10})(\overline{c_2} + c_1)(\overline{c_2} + c_{11})(\overline{c_2} + c_1 + c_4) \\
& \cdot(\overline{c_3} + c_2 + c_6)(\overline{c_3} + c_4)(\overline{c_4} + c_1)(\overline{c_4} + c_1)(\overline{c_6} + c_{11})(\overline{c_6} + c_1 + c_4) \\
& \cdot(\overline{c_7} + c_2 + c_6)(\overline{c_8} + c_2 + c_6)(\overline{c_8} + c_3 + c_9)(\overline{c_9} + c_4) \quad = \quad 1
\end{aligned}
$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{ccccccccccccc}
c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
1 & - & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & 1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
- & 1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
1 & 1 & - & 1 & - & 1 & - & - & - & 1 & - & - & 4 \\
1 & - & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & - & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & - & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
1 & 0 & - & - & - & - & - & - & - & - & - & - & 9 \\
- & 0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
1 & 0 & - & 1 & - & - & - & - & - & - & - & - & 11 \\
- & 1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & - & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
1 & - & - & 0 & - & - & - & - & - & - & - & - & 14 \\
1 & - & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
1 & - & - & 1 & - & 0 & - & - & - & - & - & - & 17 \\
- & 1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
- & 1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & - & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

Solving the Covering Problem

Rows 4, 11, and 17 dominate row 5, Row 14 dominates row 15.

$$\mathbf{A} = \begin{array}{cccccccccccccc}
 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
\left[\right. & 1 & - & - & - & - & - & - & - & - & - & 1 & - & \left.\right] \quad 1 \\
 & 1 & 1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
 & - & 1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
 & 1 & 1 & - & 1 & - & 1 & - & - & - & 1 & - & - & 4 \\
 & 1 & - & - & 1 & - & - & - & - & - & - & - & - & 5 \\
 & - & - & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
 & - & - & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
 & - & - & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
 & 1 & 0 & - & - & - & - & - & - & - & - & - & - & 9 \\
 & - & 0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
 & 1 & 0 & - & 1 & - & - & - & - & - & - & - & - & 11 \\
 & - & 1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
 & - & - & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
 & 1 & - & - & 0 & - & - & - & - & - & - & - & - & 14 \\
 & 1 & - & - & 0 & - & - & - & - & - & - & - & - & 15 \\
 & - & - & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
 & 1 & - & - & 1 & - & 0 & - & - & - & - & - & - & 17 \\
 & - & 1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
 & - & 1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
 & - & - & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
\left[\right. & - & - & - & 1 & - & - & - & - & 0 & - & - & - & \left.\right] \quad 21
\end{array}$$

Solving the Covering Problem

Rows 4, 11, and 17 dominate row 5, Row 14 dominates row 15.

$$\mathbf{A} = \begin{array}{cccccccccccc|c}
c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
\hline
1 & - & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & 1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
- & 1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
1 & - & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & - & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & - & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
1 & 0 & - & - & - & - & - & - & - & - & - & - & 9 \\
- & 0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
- & 1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & - & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
1 & - & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
- & 1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
- & 1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & - & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

Solving the Covering Problem

Cyclic

$$\mathbf{A} = \begin{array}{ccccccccccccc}
c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
1 & - & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & 1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
- & 1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
1 & - & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & - & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & - & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
1 & 0 & - & - & - & - & - & - & - & - & - & - & 9 \\
- & 0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
- & 1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & - & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
1 & - & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
- & 1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
- & 1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & - & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

Solving the Covering Problem

MIS = {1, 6, 8}, so $L = 3$

$$\mathbf{A} = \begin{array}{ccccccccccccc}
c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
1 & - & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & 1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
- & 1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
1 & - & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & - & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & - & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
1 & 0 & - & - & - & - & - & - & - & - & - & - & 9 \\
- & 0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
- & 1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & - & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
1 & - & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
- & 1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
- & 1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & - & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

Solving the Covering Problem

c_1 has a branching weight of 1.33 which is best.

$$\mathbf{A} = \begin{array}{cccccccccccc|c}
c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
\hline
1 & - & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & 1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
- & 1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
1 & - & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & - & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & - & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
1 & 0 & - & - & - & - & - & - & - & - & - & - & 9 \\
- & 0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
- & 1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & - & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
1 & - & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
- & 1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
- & 1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & - & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{ccccccccccc c}
c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
- & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
- & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

$$c_1 = 1$$

Solving the Covering Problem

Column c_4 dominates c_{10}.

$$\mathbf{A} = \begin{array}{ccccccccccc|c}
c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
- & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
- & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

$$c_1 = 1$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cccccccccc|c}
c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & c_{12} & \\
\hline
1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & 3 \\
- & 1 & - & - & - & 1 & - & 1 & - & 1 & 6 \\
- & 1 & - & - & - & - & 1 & 1 & 1 & - & 7 \\
- & - & 1 & - & - & - & - & - & - & - & 8 \\
0 & - & - & - & - & - & - & - & 1 & - & 10 \\
1 & 0 & - & - & 1 & - & - & - & - & - & 12 \\
- & 0 & 1 & - & - & - & - & - & - & - & 13 \\
- & - & - & - & 0 & - & - & - & 1 & - & 16 \\
1 & - & - & - & 1 & 0 & - & - & - & - & 18 \\
1 & - & - & - & 1 & - & 0 & - & - & - & 19 \\
- & 1 & - & - & - & - & 0 & 1 & - & - & 20 \\
- & - & 1 & - & - & - & - & 0 & - & - & 21
\end{array}$$

$$c_1 = 1, c_{10} = 0$$

Solving the Covering Problem

c_4 essential.

$$\mathbf{A} = \begin{array}{cccccccccc|c}
c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & c_{12} & \\
\hline
1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & 3 \\
- & 1 & - & - & - & 1 & - & 1 & - & 1 & 6 \\
- & 1 & - & - & - & - & 1 & 1 & 1 & - & 7 \\
- & - & 1 & - & - & - & - & - & - & - & 8 \\
0 & - & - & - & - & - & - & - & 1 & - & 10 \\
1 & 0 & - & - & 1 & - & - & - & - & - & 12 \\
- & 0 & 1 & - & - & - & - & - & - & - & 13 \\
- & - & - & - & 0 & - & - & - & 1 & - & 16 \\
1 & - & - & - & 1 & 0 & - & - & - & - & 18 \\
1 & - & - & - & 1 & - & 0 & - & - & - & 19 \\
- & 1 & - & - & - & - & 0 & 1 & - & - & 20 \\
- & - & 1 & - & - & - & - & 0 & - & - & 21
\end{array}$$

$$c_1 = 1, c_{10} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cccccccccc} c_2 & c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & c_{12} & \\ 1 & 1 & 1 & 1 & 1 & 1 & - & - & - & 3 \\ - & 1 & - & - & 1 & - & 1 & - & 1 & 6 \\ - & 1 & - & - & - & 1 & 1 & 1 & - & 7 \\ 0 & - & - & - & - & - & - & 1 & - & 10 \\ 1 & 0 & - & 1 & - & - & - & - & - & 12 \\ - & - & - & 0 & - & - & - & 1 & - & 16 \\ 1 & - & - & 1 & 0 & - & - & - & - & 18 \\ 1 & - & - & 1 & - & 0 & - & - & - & 19 \\ - & 1 & - & - & - & 0 & 1 & - & - & 20 \end{array}$$

$$c_1 = 1, c_4 = 1, c_{10} = 0$$

Solving the Covering Problem

c_9 dominates c_{12}.

$$\mathbf{A} = \begin{array}{c}
\begin{array}{ccccccccc} c_2 & c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & c_{12} \end{array} & \\
\left[\begin{array}{ccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & - & - & - \\
- & 1 & - & - & 1 & - & 1 & - & 1 \\
- & 1 & - & - & - & 1 & 1 & 1 & - \\
0 & - & - & - & - & - & - & 1 & - \\
1 & 0 & - & 1 & - & - & - & - & - \\
- & - & - & 0 & - & - & - & 1 & - \\
1 & - & - & 1 & 0 & - & - & - & - \\
1 & - & - & 1 & - & 0 & - & - & - \\
- & 1 & - & - & - & 0 & 1 & - & -
\end{array}\right] & \begin{array}{r} 3 \\ 6 \\ 7 \\ 10 \\ 12 \\ 16 \\ 18 \\ 19 \\ 20 \end{array}
\end{array}$$

$$c_1 = 1, c_4 = 1, c_{10} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cccccccccc}
 & c_2 & c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\
\left[\right. & 1 & 1 & 1 & 1 & 1 & 1 & - & - & \left.\right] \quad 3 \\
 & - & 1 & - & - & 1 & - & 1 & - & \quad 6 \\
 & - & 1 & - & - & - & 1 & 1 & 1 & \quad 7 \\
 & 0 & - & - & - & - & - & - & 1 & \quad 10 \\
 & 1 & 0 & - & 1 & - & - & - & - & \quad 12 \\
 & - & - & - & 0 & - & - & - & 1 & \quad 16 \\
 & 1 & - & - & 1 & 0 & - & - & - & \quad 18 \\
 & 1 & - & - & 1 & - & 0 & - & - & \quad 19 \\
\left.\right. & - & 1 & - & - & - & 0 & 1 & - & \left.\right] \quad 20
\end{array}$$

$$c_1 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Cyclic

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccccc|c} c_2 & c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\ 1 & 1 & 1 & 1 & 1 & 1 & - & - & 3 \\ - & 1 & - & - & 1 & - & 1 & - & 6 \\ - & 1 & - & - & - & 1 & 1 & 1 & 7 \\ 0 & - & - & - & - & - & - & 1 & 10 \\ 1 & 0 & - & 1 & - & - & - & - & 12 \\ - & - & - & 0 & - & - & - & 1 & 16 \\ 1 & - & - & 1 & 0 & - & - & - & 18 \\ 1 & - & - & 1 & - & 0 & - & - & 19 \\ - & 1 & - & - & - & 0 & 1 & - & 20 \end{array}$$

$$c_1 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

MIS = { 3 }, $L = 3$

$$\mathbf{A} = \begin{array}{ccccccccc} c_2 & c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\ 1 & 1 & 1 & 1 & 1 & 1 & - & - & 3 \\ - & 1 & - & - & 1 & - & 1 & - & 6 \\ - & 1 & - & - & - & 1 & 1 & 1 & 7 \\ 0 & - & - & - & - & - & - & 1 & 10 \\ 1 & 0 & - & 1 & - & - & - & - & 12 \\ - & - & - & 0 & - & - & - & 1 & 16 \\ 1 & - & - & 1 & 0 & - & - & - & 18 \\ 1 & - & - & 1 & - & 0 & - & - & 19 \\ - & 1 & - & - & - & 0 & 1 & - & 20 \end{array}$$

$$c_1 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

c_3 has best branching value of 0.75.

$$\mathbf{A} = \begin{array}{cc} \begin{array}{cccccccc} c_2 & c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} \end{array} & \\ \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & - & - \\ - & 1 & - & - & 1 & - & 1 & - \\ - & 1 & - & - & - & 1 & 1 & 1 \\ 0 & - & - & - & - & - & - & 1 \\ 1 & 0 & - & 1 & - & - & - & - \\ - & - & - & 0 & - & - & - & 1 \\ 1 & - & - & 1 & 0 & - & - & - \\ 1 & - & - & 1 & - & 0 & - & - \\ - & 1 & - & - & - & 0 & 1 & - \end{array}\right] & \begin{array}{r} 3 \\ 6 \\ 7 \\ 10 \\ 12 \\ 16 \\ 18 \\ 19 \\ 20 \end{array} \end{array}$$

$$c_1 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{array}{ccccccccc} & c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\ \left[\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right. & 0 & - & - & - & - & - & 1 & \left.\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right] \end{array}$$

$$\mathbf{A} = \begin{array}{c|ccccccc|c} & c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\ \hline & 0 & - & - & - & - & - & 1 & 10 \\ & 1 & - & 1 & - & - & - & - & 12 \\ & - & - & 0 & - & - & - & 1 & 16 \\ & 1 & - & 1 & 0 & - & - & - & 18 \\ & 1 & - & 1 & - & 0 & - & - & 19 \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Rows 18 and 19 dominate row 12.

$$\mathbf{A} = \begin{array}{cc} \begin{array}{ccccccc} c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} \end{array} & \\ \left[\begin{array}{ccccccc} 0 & - & - & - & - & - & 1 \\ 1 & - & 1 & - & - & - & - \\ - & - & 0 & - & - & - & 1 \\ 1 & - & 1 & 0 & - & - & - \\ 1 & - & 1 & - & 0 & - & - \end{array}\right] & \begin{array}{c} 10 \\ 12 \\ 16 \\ 18 \\ 19 \end{array} \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cc} \begin{array}{ccccccc} c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} \end{array} & \\ \left[\begin{array}{ccccccc} 0 & - & - & - & - & - & 1 \\ 1 & - & 1 & - & - & - & - \\ - & - & 0 & - & - & - & 1 \end{array}\right] & \begin{array}{c} 10 \\ 12 \\ 16 \end{array} \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Column c_{11} dominates c_5, c_7, c_8, and c_9.

$$\mathbf{A} = \begin{array}{c|ccccccc|c} & c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\ & 0 & - & - & - & - & - & 1 & 10 \\ & 1 & - & 1 & - & - & - & - & 12 \\ & - & - & 0 & - & - & - & 1 & 16 \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{ccc} c_2 & c_6 & c_{11} \\ \left[\begin{array}{ccc} 0 & - & 1 \\ 1 & 1 & - \\ - & 0 & 1 \end{array}\right. & & \end{array}$$

$$\mathbf{A} = \begin{array}{ccccc} & c_2 & c_6 & c_{11} & \\ \hline & 0 & - & 1 & 10 \\ & 1 & 1 & - & 12 \\ & - & 0 & 1 & 16 \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Cyclic

$$\mathbf{A} = \begin{array}{c c c c} c_2 & c_6 & c_{11} & \\ 0 & - & 1 & 10 \\ 1 & 1 & - & 12 \\ - & 0 & 1 & 16 \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

MIS = { 12 }, $L = 4$.

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{ccc} c_2 & c_6 & c_{11} \end{array}$$

$$\mathbf{A} = \begin{array}{cccc} c_2 & c_6 & c_{11} & \\ \left[\begin{array}{c} 0 \\ 1 \\ - \end{array}\right. & \begin{array}{c} - \\ 1 \\ 0 \end{array} & \left.\begin{array}{c} 1 \\ - \\ 1 \end{array}\right] & \begin{array}{c} 10 \\ 12 \\ 16 \end{array} \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Branch on c_2.

$$\mathbf{A} = \begin{array}{cc} & \begin{array}{ccc} c_2 & c_6 & c_{11} \end{array} \\ & \left[\begin{array}{ccc} 0 & - & 1 \\ 1 & 1 & - \\ - & 0 & 1 \end{array} \right] \begin{array}{c} 10 \\ 12 \\ 16 \end{array} \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{c} \\ \\ \end{array} \begin{array}{cc|c} c_6 & c_{11} & \\ - & 1 & 10 \\ 0 & 1 & 16 \end{array}$$

$$c_1 = 1, c_2 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

c_{11} is essential.

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \end{array} \begin{array}{cc} c_6 & c_{11} \\ \left[\begin{array}{cc} - & 1 \\ 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \\ 10 \\ 16 \end{array}$$

$$c_1 = 1, c_2 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Solution is $\{c_1, c_2, c_3, c_4, c_{11}\}$.

This is best solution so far, but cost of 5 is greater than lower bound of 4.

$$\mathbf{A} = \begin{array}{c} \\ \\ \end{array}\begin{array}{cc|c} c_6 & c_{11} & \\ \end{array}$$

$$\mathbf{A} = \begin{array}{c} \\ \end{array}\begin{array}{cc} c_6 & c_{11} \\ \left[\begin{array}{c} - \\ 0 \end{array}\right. & \left.\begin{array}{c} 1 \\ 1 \end{array}\right] \end{array} \begin{array}{c} 10 \\ 16 \end{array}$$

$c_1 = 1, c_2 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$

Solving the Covering Problem

Let's try $c_2 = 0$.

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{cccc} c_2 & c_6 & c_{11} & \\ \end{array}$$

$$\mathbf{A} = \begin{array}{c|ccc|c} & c_2 & c_6 & c_{11} & \\ \hline & 0 & - & 1 & 10 \\ & 1 & 1 & - & 12 \\ & - & 0 & 1 & 16 \end{array}$$

$$c_1 = 1, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{ccc} c_6 & c_{11} & \\ \left[\begin{array}{c} 1 \\ 0 \end{array}\right. & \left.\begin{array}{c} - \\ 1 \end{array}\right] & \begin{array}{c} 12 \\ 16 \end{array} \end{array}$$

$$c_1 = 1, c_2 = 0, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Now must select both c_6 and c_{11}, so another solution of cost 5.

$$\mathbf{A} = \begin{array}{c} \\ \\ \end{array} \begin{array}{cc} c_6 & c_{11} \\ \left[\begin{array}{c} 1 \\ 0 \end{array}\right. & \left.\begin{array}{c} - \\ 1 \end{array}\right] \end{array} \begin{array}{c} \\ 12 \\ 16 \end{array}$$

$$c_1 = 1, c_2 = 0, c_3 = 1, c_4 = 1, c_5 = 0, c_7 = 0, c_8 = 0, c_9 = 0, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Let's go back and try c_3 equal to 0.

$$\mathbf{A} = \begin{array}{cccccccccc}
 & c_2 & c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\
\lceil & 1 & 1 & 1 & 1 & 1 & 1 & - & - & 3 \\
| & - & 1 & - & - & 1 & - & 1 & - & 6 \\
| & - & 1 & - & - & - & 1 & 1 & 1 & 7 \\
| & 0 & - & - & - & - & - & - & 1 & 10 \\
| & 1 & 0 & - & 1 & - & - & - & - & 12 \\
| & - & - & - & 0 & - & - & - & 1 & 16 \\
| & 1 & - & - & 1 & 0 & - & - & - & 18 \\
| & 1 & - & - & 1 & - & 0 & - & - & 19 \\
\lfloor & - & 1 & - & - & - & 0 & 1 & - & 20
\end{array}$$

$$c_1 = 1, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccccc} c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\ 1 & 1 & 1 & 1 & 1 & - & - & 3 \\ - & - & - & 1 & - & 1 & - & 6 \\ - & - & - & - & 1 & 1 & 1 & 7 \\ 0 & - & - & - & - & - & 1 & 10 \\ - & - & 0 & - & - & - & 1 & 16 \\ 1 & - & 1 & 0 & - & - & - & 18 \\ 1 & - & 1 & - & 0 & - & - & 19 \\ - & - & - & - & 0 & 1 & - & 20 \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Cyclic

$$\mathbf{A} = \begin{array}{c c c c c c c c}
c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\
\end{array}$$

$$\mathbf{A} = \begin{array}{cc} \begin{array}{ccccccc} c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} \end{array} & \\ \left[\begin{array}{ccccccc}
1 & 1 & 1 & 1 & 1 & - & - \\
- & - & - & 1 & - & 1 & - \\
- & - & - & - & 1 & 1 & 1 \\
0 & - & - & - & - & - & 1 \\
- & - & 0 & - & - & - & 1 \\
1 & - & 1 & 0 & - & - & - \\
1 & - & 1 & - & 0 & - & - \\
- & - & - & - & 0 & 1 & -
\end{array}\right] & \begin{array}{r} 3 \\ 6 \\ 7 \\ 10 \\ 16 \\ 18 \\ 19 \\ 20 \end{array} \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Branch on c_9.

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{ccccccc|c} c_2 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{11} & \\ 1 & 1 & 1 & 1 & 1 & - & - & 3 \\ - & - & - & 1 & - & 1 & - & 6 \\ - & - & - & - & 1 & 1 & 1 & 7 \\ 0 & - & - & - & - & - & 1 & 10 \\ - & - & 0 & - & - & - & 1 & 16 \\ 1 & - & 1 & 0 & - & - & - & 18 \\ 1 & - & 1 & - & 0 & - & - & 19 \\ - & - & - & - & 0 & 1 & - & 20 \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{c c} \begin{array}{cccccc} c_2 & c_5 & c_6 & c_7 & c_8 & c_{11} \end{array} & \\ \left[\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & - \\ 0 & - & - & - & - & 1 \\ - & - & 0 & - & - & 1 \\ 1 & - & 1 & 0 & - & - \\ 1 & - & 1 & - & 0 & - \end{array}\right] & \begin{array}{r} 3 \\ 10 \\ 16 \\ 18 \\ 19 \end{array} \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_9 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Column c_5 dominates c_7 and c_8.

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccc|c} c_2 & c_5 & c_6 & c_7 & c_8 & c_{11} & \\ 1 & 1 & 1 & 1 & 1 & - & 3 \\ 0 & - & - & - & - & 1 & 10 \\ - & - & 0 & - & - & 1 & 16 \\ 1 & - & 1 & 0 & - & - & 18 \\ 1 & - & 1 & - & 0 & - & 19 \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_9 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{c|cccc|c} & c_2 & c_5 & c_6 & c_{11} & \\ & 1 & 1 & 1 & - & 3 \\ & 0 & - & - & 1 & 10 \\ & - & - & 0 & 1 & 16 \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_7 = 0, c_8 = 0, c_9 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Column c_5 dominates c_2 and c_6.

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{cccc} c_2 & c_5 & c_6 & c_{11} \\ \end{array}$$

$$\mathbf{A} = \begin{array}{cc} \begin{array}{cccc} c_2 & c_5 & c_6 & c_{11} \end{array} & \\ \left[\begin{array}{cccc} 1 & 1 & 1 & - \\ 0 & - & - & 1 \\ - & - & 0 & 1 \end{array}\right] & \begin{array}{r} 3 \\ 10 \\ 16 \end{array} \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_7 = 0, c_8 = 0, c_9 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cc} c_5 & c_{11} \\ \left[\begin{array}{cc} 1 & - \end{array}\right] & \end{array} \quad 3$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_7 = 0, c_8 = 0, c_9 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

c_5 is essential.

$$\mathbf{A} = \begin{array}{c} \\ \\ \end{array} \begin{array}{cc} c_5 & c_{11} \\ \left[1 \right. & \left. - \right] \end{array} \quad \begin{array}{c} \\ 3 \end{array}$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_7 = 0, c_8 = 0, c_9 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Found solution $\{c_1, c_4, c_5, c_9\}$ with cost 4.

Not as good as lower bound of 3.

Continue with $c_9 = 0$ to obtain solution $\{c_1, c_2, c_4, c_7, c_{11}\}$.

$$\mathbf{A} = \begin{array}{c} \\ \end{array} \begin{array}{cc} c_5 & c_{11} \\ \left[\begin{array}{c} 1 \end{array} \right. & \left. \begin{array}{c} - \end{array} \right] \end{array} \quad 3$$

$$c_1 = 1, c_3 = 0, c_4 = 1, c_7 = 0, c_8 = 0, c_9 = 1, c_{10} = 0, c_{12} = 0$$

Solving the Covering Problem

Let's try $c_1 = 0$.

$$\mathbf{A} = \begin{array}{cccccccccccc|c}
c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
\hline
1 & - & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & 1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
- & 1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
1 & - & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & - & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & - & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
1 & 0 & - & - & - & - & - & - & - & - & - & - & 9 \\
- & 0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
- & 1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & - & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
1 & - & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
- & 1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
- & 1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & - & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cccccccccccc}
c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
- & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
- & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
0 & - & - & - & - & - & - & - & - & - & - & 9 \\
0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
- & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}$$

$$c_1 = 0$$

Solving the Covering Problem

c_{11} is essential and c_2 and c_4 are unacceptable.

$$
\mathbf{A} = \begin{array}{ccccccccccc|c}
c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & \\
- & - & - & - & - & - & - & - & - & 1 & - & 1 \\
1 & - & - & 1 & - & - & - & - & - & - & - & 2 \\
1 & 1 & - & 1 & 1 & 1 & 1 & - & - & - & - & 3 \\
- & - & 1 & - & - & - & - & - & - & - & - & 5 \\
- & 1 & - & - & - & 1 & - & 1 & - & - & 1 & 6 \\
- & 1 & - & - & - & - & 1 & 1 & - & 1 & - & 7 \\
- & - & 1 & - & - & - & - & - & 1 & - & - & 8 \\
0 & - & - & - & - & - & - & - & - & - & - & 9 \\
0 & - & - & - & - & - & - & - & - & 1 & - & 10 \\
1 & 0 & - & - & 1 & - & - & - & - & - & - & 12 \\
- & 0 & 1 & - & - & - & - & - & - & - & - & 13 \\
- & - & 0 & - & - & - & - & - & - & - & - & 15 \\
- & - & - & - & 0 & - & - & - & - & 1 & - & 16 \\
1 & - & - & - & 1 & 0 & - & - & - & - & - & 18 \\
1 & - & - & - & 1 & - & 0 & - & - & - & - & 19 \\
- & 1 & - & - & - & - & 0 & 1 & - & - & - & 20 \\
- & - & 1 & - & - & - & - & 0 & - & - & - & 21
\end{array}
$$

$c_1 = 0$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cccccccc|c} c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{12} & \\ - & 1 & - & - & - & - & - & - & 2 \\ 1 & 1 & 1 & 1 & 1 & - & - & - & 3 \\ - & - & - & - & - & - & - & - & 5 \\ 1 & - & - & 1 & - & 1 & - & 1 & 6 \\ - & - & - & - & - & - & 1 & - & 8 \\ 0 & - & 1 & - & - & - & - & - & 12 \\ 0 & - & - & - & - & - & - & - & 13 \\ - & - & 1 & 0 & - & - & - & - & 18 \\ - & - & 1 & - & 0 & - & - & - & 19 \\ 1 & - & - & - & 0 & 1 & - & - & 20 \\ - & - & - & - & - & 0 & - & - & 21 \end{array}$$

$$c_2 = 0, c_4 = 0, c_{11} = 1$$

Solving the Covering Problem

All rows dominate row 5.

$$\mathbf{A} = \begin{array}{cc} \begin{array}{cccccccc} c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{12} \end{array} & \\ \left[\begin{array}{cccccccc} - & 1 & - & - & - & - & - & - \\ 1 & 1 & 1 & 1 & 1 & - & - & - \\ - & - & - & - & - & - & - & - \\ 1 & - & - & 1 & - & 1 & - & 1 \\ - & - & - & - & - & - & 1 & - \\ 0 & - & 1 & - & - & - & - & - \\ 0 & - & - & - & - & - & - & - \\ - & - & 1 & 0 & - & - & - & - \\ - & - & 1 & - & 0 & - & - & - \\ 1 & - & - & - & 0 & 1 & - & - \\ - & - & - & - & - & 0 & - & - \end{array}\right] & \begin{array}{r} 2 \\ 3 \\ 5 \\ 6 \\ 8 \\ 12 \\ 13 \\ 18 \\ 19 \\ 20 \\ 21 \end{array} \end{array}$$

$$c_2 = 0, c_4 = 0, c_{11} = 1$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{c} \\ \end{array} \begin{array}{cccccccc} c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{12} \\ \left[- \right. & - & - & - & - & - & - & \left. - \right] \end{array} \quad 5$$

$$c_1 = c_2 = c_4 = 0, c_{11} = 1$$

Solving the Covering Problem

All columns mutually dominate.

$$\mathbf{A} = \begin{array}{c} \\ \\ \end{array} \begin{array}{cccccccc} c_3 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{12} \\ \left[- \right. & - & - & - & - & - & - & \left. - \right] \end{array} \quad 5$$

$$c_1 = c_2 = c_4 = 0, c_{11} = 1$$

Solving the Covering Problem

$$\mathbf{A} = \begin{array}{cc} c_3 & \\ \left[\begin{array}{c} - \end{array}\right] & 5 \end{array}$$

$$c_1 = c_2 = c_4 = c_5 = c_6 = c_7 = c_8 = c_9 = c_{10} = 0, c_{11} = 1, c_{12} = 0$$

Solving the Covering Problem

No solution, so *bcp* returns best solution of $\{c_1, c_4, c_5, c_9\}$.

$$\mathbf{A} = \begin{array}{cc} c_3 & \\ \left[\begin{array}{c} - \end{array}\right] & 5 \end{array}$$

$$c_1 = c_2 = c_4 = c_5 = c_6 = c_7 = c_8 = c_9 = c_{10} = 0, c_{11} = 1, c_{12} = 0$$

Final Solution

	Prime compatibles	Class set
1	*abde*	$\emptyset$
4	*deh*	$\{(a,b),(a,d)\}$
5	*bc*	$\emptyset$
9	*fg*	$\{(e,h)\}$

Example Huffman Flow Table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
a	a,0	–	d,0	e,1	b,0	a,–	–
b	b,0	d,1	a,–	–	a,–	a,1	–
c	b,0	d,1	a,1	–	–	–	g,0
d	–	e,–	–	b,–	b,0	–	a,–
e	b,–	e,–	a,–	–	b,–	e,–	a,1
f	b,0	c,–	–,1	h,1	f,1	g,0	–
g	–	c,1	–	e,1	–	g,0	f,0
h	a,1	e,0	d,1	b,0	b,–	e,–	a,1

Reduced Flow Table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1,0	{1,4},1	1,0	1,1	1,0	1,1	1,1
4	1,1	{1,4},0	1,1	{1,5},0	{1,5},0	{1,4},–	1,1
5	{1,5},0	{1,4},1	1,1	–	1,–	1,1	9,0
9	{1,5},0	5,1	–,1	4,1	9,1	9,0	9,0

Final Reduced Flow Table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1,0	1,1	1,0	1,1	1,0	1,1	1,1
4	1,1	1,0	1,1	1,0	1,0	1,–	1,1
5	1,0	1,1	1,1	–	1,–	1,1	9,0
9	1,0	5,1	–,1	4,1	9,1	9,0	9,0

State Assignment

- Each row must be encoded using a unique binary code.
- In synchronous design, a correct encoding can be assigned arbitrarily using n bits for a flow table with 2^n rows or less.
- In asynchronous design, more care must be taken to ensure that a circuit can be built that is independent of signal delays.

Critical Races

- When present state equals next state, circuit is *stable*.
- When codes differ in one bit, the circuit is *in transition*.
- When the codes differ in multiple bits, the circuit is *racing*.
- A race is *critical* when differences in delay can cause it to reach different stable states.
- A state assignment is correct when it is free of critical races.

Minimum Transition Time State Assignment

- A transition from state s_i to state s_j is *direct* (denoted [s_i,s_j]) when all state variables are excited to change at the same time.
- $[s_i, s_j]$ races critically with [s_k,s_l] when unequal delays can cause these transitions to pass through a common state.
- When all state transitions are direct, the state assignment is called a *minimum transition time state assignment*.
- A flow table in which each unstable state leads directly to a stable state is called a *normal flow table*.

A Simple Huffman Flow Table

	x_1	x_2	x_3	x_4
a	a	b	d	c
b	c	b	b	b
c	c	d	b	c
d	a	d	d	b

	$y_1 y_2$	$y_1 y_2 y_3$
a	00	000
b	01	011
c	10	110
d	11	101

Partition Theory

- A partition π on a set S is a set of subsets of S such that their pairwise intersection is empty.
- The disjoint subsets of π are called *blocks*.
- A partition is *completely specified* if union of subsets is S.
- Otherwise, the partition is *incompletely specified*.
- Elements of S which do not appear in π are *unspecified*.

Partition Theory and State Assignment

- n state variables $y_1, \ldots, y_n$ induce τ-*partitions* $\tau_1, \ldots, \tau_n$.
- States with $y_1 = 0$ are in one block of τ_1 while those with $y_1 = 1$ are in the other block.
- Each partition is composed of only one or two blocks.
- Order blocks appear or which is assigned a 0 or 1 is arbitrary.
- Once we find one valid assignment, others can be found by complementing or reordering variables.

Partition Example

	$y_1 y_2$	$y_1 y_2 y_3$
a	00	000
b	01	011
c	10	110
d	11	101

Partition Example

	$y_1 y_2$	$y_1 y_2 y_3$
a	00	000
b	01	011
c	10	110
d	11	101

$$\begin{aligned} \tau_1 &= \{ab; cd\} \\ \tau_2 &= \{ac; bd\} \end{aligned}$$

$$\begin{aligned} \tau_1 &= \{ab; cd\} \\ \tau_2 &= \{ad; bc\} \\ \tau_3 &= \{ac; bd\} \end{aligned}$$

Partition List

- $\pi_2 \leq \pi_1$ iff all elements specified in π_2 are specified in π_1 and each block of π_2 appears in a unique block of π_1.
- A *partition list* is a collection of partitions of the form:
  - $\{ s_p, s_q; s_r, s_s \}$ where $[s_p, s_q]$ and $[s_r, s_s]$ are transitions in the same column.
  - $\{ s_p, s_q; s_t \}$ where $[s_p, s_q]$ and is a transition in the same column as the stable state s_t.
- A state assignment for a normal flow table is a minimum transition time assignment free of critical races iff each partition in the partition list is $\leq$ some τ_i.

Tracey's Theorem

Theorem 5.2 (Tracey, 1966) A row assignment allotting one y-state per row can be used for direct transition realization of normal flow tables without critical races if, and only if, for every transition $[s_i, s_j]$:

1. If $[s_m, s_n]$ is another transition in the same column, then at least one y-variable partitions the pair $\{s_i, s_j\}$ and the pair $\{s_m, s_n\}$ into separate blocks.
2. If s_k is a stable state in the same column then at least one y-variable partitions the pair $\{s_i, s_j\}$ and the state s_k into separate blocks.
3. For $i \neq j$, s_i and s_j are in separate blocks of at least one y-variable partition.

Partition List Example

	x_1	x_2	x_3	x_4
a	a	b	d	c
b	c	b	b	b
c	c	d	b	c
d	a	d	d	b

Partition List Example

	x_1	x_2	x_3	x_4
a	a	b	d	c
b	c	b	b	b
c	c	d	b	c
d	a	d	d	b

$$\begin{array}{lcl} \pi_1 & = & \{ad;bc\} \\ \pi_2 & = & \{ab;cd\} \\ \pi_3 & = & \{ad;bc\} \\ \pi_4 & = & \{ac;bd\} \end{array}$$

Partition List Example

	x_1	x_2	x_3	x_4
a	a	b	d	c
b	c	b	b	b
c	c	d	b	c
d	a	d	d	b

$$\begin{aligned}\pi_1 &= \{ad; bc\}\\ \pi_2 &= \{ab; cd\}\\ \pi_3 &= \{ad; bc\}\\ \pi_4 &= \{ac; bd\}\end{aligned}$$

	$y_1 y_2$	$y_1 y_2 y_3$
a	00	000
b	01	011
c	10	110
d	11	101

$$\begin{aligned}\tau_1 &= \{ab; cd\}\\ \tau_2 &= \{ac; bd\}\end{aligned}$$

$$\begin{aligned}\tau_1 &= \{ab; cd\}\\ \tau_2 &= \{ad; bc\}\\ \tau_3 &= \{ac; bd\}\end{aligned}$$

Larger Example

	x_1	x_2	x_3	x_4
a	a,0	c,1	d,0	c,1
b	a,0	f,1	c,1	b,0
c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

Larger Example

	x_1	x_2	x_3	x_4
a	a,0	c,1	d,0	c,1
b	a,0	f,1	c,1	b,0
c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_1 &= \{ab;cf\} \\
\pi_2 &= \{ae;cf\} \\
\pi_3 &= \{ac;de\} \\
\pi_4 &= \{ac;bf\} \\
\pi_5 &= \{bf;de\} \\
\pi_6 &= \{ad;bc\} \\
\pi_7 &= \{ad;ce\} \\
\pi_8 &= \{ac;bd\} \\
\pi_9 &= \{ac;ef\} \\
\pi_{10} &= \{bd;ef\}
\end{aligned}$$

Boolean Matrix Example

$\pi_1 = \{ab; cf\}$
$\pi_2 = \{ae; cf\}$
$\pi_3 = \{ac; de\}$
$\pi_4 = \{ac; bf\}$
$\pi_5 = \{bf; de\}$
$\pi_6 = \{ad; bc\}$
$\pi_7 = \{ad; ce\}$
$\pi_8 = \{ac; bd\}$
$\pi_9 = \{ac; ef\}$
$\pi_{10} = \{bd; ef\}$

$$\begin{array}{ccccccc}
 & a & b & c & d & e & f \\
\pi_1 & 0 & 0 & 1 & - & - & 1 \\
\pi_2 & 0 & - & 1 & - & 0 & 1 \\
\pi_3 & 0 & - & 0 & 1 & 1 & - \\
\pi_4 & 0 & 1 & 0 & - & - & 1 \\
\pi_5 & - & 0 & - & 1 & 1 & 0 \\
\pi_6 & 0 & 1 & 1 & 0 & - & - \\
\pi_7 & 0 & - & 1 & 0 & 1 & - \\
\pi_8 & 0 & 1 & 0 & 1 & - & - \\
\pi_9 & 0 & - & 0 & - & 1 & 1 \\
\pi_{10} & - & 0 & - & 0 & 1 & 1
\end{array}$$

Boolean Matrix and State Assignment

- State assignment problem is to find a Boolean matrix C with a minimum number of rows such that each row in the original partition list matrix is covered by some row of C.
- The rows of this reduced matrix represent the τ-partitions.
- The columns of this matrix represent a state assignment.
- Number of rows is the same as the number of state variables.

Intersection

- Two rows of a Boolean matrix, R_i and R_j, have an *intersection* if R_i and R_j agree wherever both R_i and R_j are specified.
- The intersection is formed by creating a row which has specified values taken from either R_i or R_j.
- Entries where neither R_i or R_j are specified are left unspecified.
- A row, R_i, *includes* another row, R_j, when R_j agrees with R_i wherever R_i is specified.
- A row, R_i, *covers* another row, R_j, if R_j includes R_i or R_j includes the complement of R_i.
- The complement of R_i is denoted $\overline{R_i}$.

Boolean Matrix Reduction

$$\begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
\pi_1 & 0 & 0 & 1 & - & - & 1 \\
\pi_2 & 0 & - & 1 & - & 0 & 1 \\
\pi_3 & 0 & - & 0 & 1 & 1 & - \\
\pi_4 & 0 & 1 & 0 & - & - & 1 \\
\pi_5 & - & 0 & - & 1 & 1 & 0 \\
\pi_6 & 0 & 1 & 1 & 0 & - & - \\
\pi_7 & 0 & - & 1 & 0 & 1 & - \\
\pi_8 & 0 & 1 & 0 & 1 & - & - \\
\pi_9 & 0 & - & 0 & - & 1 & 1 \\
\pi_{10} & - & 0 & - & 0 & 1 & 1
\end{array}$$

Boolean Matrix Reduction

$$\begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
(\pi_1,\pi_2) & 0 & 0 & 1 & - & 0 & 1 \\
\pi_3 & 0 & - & 0 & 1 & 1 & - \\
\pi_4 & 0 & 1 & 0 & - & - & 1 \\
\pi_5 & - & 0 & - & 1 & 1 & 0 \\
\pi_6 & 0 & 1 & 1 & 0 & - & - \\
\pi_7 & 0 & - & 1 & 0 & 1 & - \\
\pi_8 & 0 & 1 & 0 & 1 & - & - \\
\pi_9 & 0 & - & 0 & - & 1 & 1 \\
\pi_{10} & - & 0 & - & 0 & 1 & 1
\end{array}$$

Boolean Matrix Reduction

$$\begin{array}{ccccccc}
 & a & b & c & d & e & f \\
(\pi_1,\pi_2) & 0 & 0 & 1 & - & 0 & 1 \\
(\pi_3,\pi_4) & 0 & 1 & 0 & 1 & 1 & 1 \\
\pi_5 & - & 0 & - & 1 & 1 & 0 \\
\pi_6 & 0 & 1 & 1 & 0 & - & - \\
\pi_7 & 0 & - & 1 & 0 & 1 & - \\
\pi_8 & 0 & 1 & 0 & 1 & - & - \\
\pi_9 & 0 & - & 0 & - & 1 & 1 \\
\pi_{10} & - & 0 & - & 0 & 1 & 1
\end{array}$$

Boolean Matrix Reduction

$$\begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
(\pi_1,\pi_2) & 0 & 0 & 1 & - & 0 & 1 \\
(\pi_3,\pi_4,\pi_8) & 0 & 1 & 0 & 1 & 1 & 1 \\
\pi_5 & - & 0 & - & 1 & 1 & 0 \\
\pi_6 & 0 & 1 & 1 & 0 & - & - \\
\pi_7 & 0 & - & 1 & 0 & 1 & - \\
\pi_9 & 0 & - & 0 & - & 1 & 1 \\
\pi_{10} & - & 0 & - & 0 & 1 & 1
\end{array}$$

Boolean Matrix Reduction

	a	b	c	d	e	f
(π_1,π_2)	0	0	1	–	0	1
$(\pi_3,\pi_4,\pi_8,\pi_9)$	0	1	0	1	1	1
π_5	–	0	–	1	1	0
π_6	0	1	1	0	–	–
π_7	0	–	1	0	1	–
π_{10}	–	0	–	0	1	1

Boolean Matrix Reduction

$$
\begin{array}{lcccccc}
 & a & b & c & d & e & f \\
(\pi_1,\pi_2) & 0 & 0 & 1 & - & 0 & 1 \\
(\pi_3,\pi_4,\pi_8,\pi_9) & 0 & 1 & 0 & 1 & 1 & 1 \\
(\overline{\pi_5},\pi_6) & 0 & 1 & 1 & 0 & 0 & 1 \\
\pi_7 & 0 & - & 1 & 0 & 1 & - \\
\pi_{10} & - & 0 & - & 0 & 1 & 1
\end{array}
$$

Boolean Matrix Reduction

$$\begin{array}{lcccccc}
 & a & b & c & d & e & f \\
(\pi_1,\pi_2) & 0 & 0 & 1 & - & 0 & 1 \\
(\pi_3,\pi_4,\pi_8,\pi_9) & 0 & 1 & 0 & 1 & 1 & 1 \\
(\overline{\pi_5},\pi_6) & 0 & 1 & 1 & 0 & 0 & 1 \\
(\pi_7,\pi_{10}) & 0 & 0 & 1 & 0 & 1 & 1
\end{array}$$

Boolean Matrix Reduction

$$\begin{array}{rcccccc}
 & a & b & c & d & e & f \\
(\pi_1,\pi_2) & 0 & 0 & 1 & - & 0 & 1 \\
(\pi_3,\pi_4,\pi_8,\pi_9) & 0 & 1 & 0 & 1 & 1 & 1 \\
(\overline{\pi_5},\pi_6) & 0 & 1 & 1 & 0 & 0 & 1 \\
(\pi_7,\pi_{10}) & 0 & 0 & 1 & 0 & 1 & 1
\end{array}$$

	$y_1 y_2 y_3 y_4$
a	0000
b	0110
c	1011
d	-100
e	0101
f	1111

Minimal Boolean Matrix

$$\begin{array}{lcccccc}
 & \text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f} \\
(\pi_1,\pi_7,\pi_{10}) & 0 & 0 & 1 & 0 & 1 & 1 \\
(\pi_2,\overline{\pi_5},\pi_6) & 0 & 1 & 1 & 0 & 0 & 1 \\
(\pi_3,\pi_4,\pi_8,\pi_9) & 0 & 1 & 0 & 1 & 1 & 1
\end{array}$$

	$y_1 y_2 y_3$
a	000
b	011
c	110
d	001
e	101
f	111

Intersectables

- If a set of rows, $\pi_i, \pi_j, \ldots, \pi_k$, have an intersection, they are called an *intersectable*.
- An intersectable may be enlarged by adding a row π_l iff π_l has an intersection with every element in the set.
- An intersectable which cannot be enlarged further is called a *maximal intersectable*.

Finding Pairwise Intersectables

- For each pair for rows, R_i and R_j, check whether R_i and R_j have an intersection.
- Also must check whether R_i and $\overline{R_j}$ have an intersection.
- If there are n partitions to cover, this implies the need to consider $2n$ *ordered partitions*.

  Theorem 5.3 (Unger, 1969) Let D be a set of ordered partitions derived from some set of unordered partitions. For some state s, label as p_1, p_2, etc. the members of D having s in their left sets, and label as q_1, q_2, etc. the members of D that do not contain s in either set. Then a minimal set of maximal intersectibles covering each member of D or its complement can be found by considering only the ordered partitions labeled as p's or q's. (The complements of the p's can be ignored.)

Finding Pairwise Intersectables

	a	b	c	d	e	f
π_1	0	0	1	–	–	1
π_2	0	–	1	–	0	1
π_3	0	–	0	1	1	–
π_4	0	1	0	–	–	1
π_5	–	0	–	1	1	0
π_6	0	1	1	0	–	–
π_7	0	–	1	0	1	–
π_8	0	1	0	1	–	–
π_9	0	–	0	–	1	1
π_{10}	–	0	–	0	1	1

Finding Pairwise Intersectables

π_2											
π_3											
π_4											
π_5											
π_6											
π_7											
π_8											
π_9											
π_{10}											
$\overline{\pi_5}$											
$\overline{\pi_{10}}$											
	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}	$\overline{\pi_5}$

Finding Pairwise Intersectables

π_2	$\sim$										
π_3	$\times$	$\times$									
π_4	$\times$	$\times$	$\sim$								
π_5	$\times$	$\times$	$\sim$	$\times$							
π_6	$\times$	$\sim$	$\times$	$\times$	$\times$						
π_7	$\sim$	$\times$	$\times$	$\times$	$\times$	$\sim$					
π_8	$\times$	$\times$	$\sim$	$\sim$	$\times$	$\times$	$\times$				
π_9	$\times$	$\times$	$\sim$	$\sim$	$\times$	$\times$	$\times$	$\sim$			
π_{10}	$\sim$	$\times$	$\times$	$\times$	$\times$	$\times$	$\sim$	$\times$	$\sim$		
$\overline{\pi_5}$	$\times$	$\sim$	$\times$	$\sim$	$\times$	$\sim$	$\times$	$\times$	$\times$	$\times$	
$\overline{\pi_{10}}$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\sim$	$\times$	$\times$	$\times$
	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}	$\overline{\pi_5}$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} =$

$S_{\pi_7} =$

$S_{\pi_6} =$

$S_{\pi_4} =$

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9, \overline{\pi_{10}}$:

$S_{\pi_7} =$

$S_{\pi_6} =$

$S_{\pi_4} =$

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9,\overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} =$

$S_{\pi_6} =$

$S_{\pi_4} =$

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9, \overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} = \pi_{10}$:

$S_{\pi_6} =$

$S_{\pi_4} =$

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step:	$c = \{(\pi_9,\pi_{10})\}$
$S_{\pi_8} = \pi_9, \overline{\pi_{10}}$:	$c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$
$S_{\pi_7} = \pi_{10}$:	$c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10})\}$
$S_{\pi_6} =$	
$S_{\pi_4} =$	
$S_{\pi_3} =$	

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9, \overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} = \pi_{10}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10})\}$

$S_{\pi_6} = \pi_7, \overline{\pi_5}$:

$S_{\pi_4} =$

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9, \overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} = \pi_{10}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10})\}$

$S_{\pi_6} = \pi_7, \overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),$
$(\pi_6,\pi_7),(\pi_6,\overline{\pi_5})\}$

$S_{\pi_4} =$

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9, \overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} = \pi_{10}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10})\}$

$S_{\pi_6} = \pi_7, \overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),$
$(\pi_6,\pi_7),(\pi_6,\overline{\pi_5})\}$

$S_{\pi_4} = \pi_8, \pi_9, \overline{\pi_5}$:

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9,\overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} = \pi_{10}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10})\}$

$S_{\pi_6} = \pi_7,\overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),(\pi_6,\overline{\pi_5})\}$

$S_{\pi_4} = \pi_8,\pi_9,\overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),(\pi_6,\overline{\pi_5}),(\pi_4,\pi_8,\pi_9),(\pi_4,\overline{\pi_5})\}$

$S_{\pi_3} =$

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9,\overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} = \pi_{10}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10})\}$

$S_{\pi_6} = \pi_7,\overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),$
$(\pi_6,\pi_7),(\pi_6,\overline{\pi_5})\}$

$S_{\pi_4} = \pi_8,\pi_9,\overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),$
$(\pi_6,\overline{\pi_5}),(\pi_4,\pi_8,\pi_9),(\pi_4,\overline{\pi_5})\}$

$S_{\pi_3} = \pi_4,\pi_5,\pi_8,\pi_9$:

Finding Maximal Intersectables

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

First step: $c = \{(\pi_9,\pi_{10})\}$

$S_{\pi_8} = \pi_9,\overline{\pi_{10}}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}})\}$

$S_{\pi_7} = \pi_{10}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10})\}$

$S_{\pi_6} = \pi_7, \overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\pi_9),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),$
$(\pi_6,\pi_7),(\pi_6,\overline{\pi_5})\}$

$S_{\pi_4} = \pi_8,\pi_9,\overline{\pi_5}$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),$
$(\pi_6,\overline{\pi_5}),(\pi_4,\pi_8,\pi_9),(\pi_4,\overline{\pi_5})\}$

$S_{\pi_3} = \pi_4,\pi_5,\pi_8,\pi_9$: $c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),$
$(\pi_6,\overline{\pi_5}),(\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),$
$(\pi_3,\pi_5)\}$

Finding Maximal Intersectables (cont)

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

$$S_{\pi_3} = \pi_4,\pi_5,\pi_8,\pi_9: \quad c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),$$
$$(\pi_6,\overline{\pi_5}),(\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),$$
$$(\pi_3,\pi_5)\}$$

$S_{\pi_2} =$

$S_{\pi_1} =$

Finding Maximal Intersectables (cont)

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

$$S_{\pi_3} = \pi_4,\pi_5,\pi_8,\pi_9: \quad c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7), (\pi_6,\overline{\pi_5}),(\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9), (\pi_3,\pi_5)\}$$

$$S_{\pi_2} = \pi_6,\overline{\pi_5}:$$

$$S_{\pi_1} =$$

Finding Maximal Intersectables (cont)

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

$$\begin{aligned}
S_{\pi_3} = \pi_4,\pi_5,\pi_8,\pi_9: \quad & c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),\\
& (\pi_6,\overline{\pi_5}),(\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),\\
& (\pi_3,\pi_5)\}\\
S_{\pi_2} = \pi_6,\overline{\pi_5}: \quad & c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),\\
& (\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),(\pi_3,\pi_5),\\
& (\pi_2,\pi_6,\overline{\pi_5})\}\\
S_{\pi_1} =
\end{aligned}$$

Finding Maximal Intersectables (cont)

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

$$\begin{aligned}
S_{\pi_3} = \pi_4,\pi_5,\pi_8,\pi_9: \quad & c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),\\
& (\pi_6,\overline{\pi_5}),(\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),\\
& (\pi_3,\pi_5)\}\\
S_{\pi_2} = \pi_6,\overline{\pi_5}: \quad & c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),\\
& (\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),(\pi_3,\pi_5),\\
& (\pi_2,\pi_6,\overline{\pi_5})\}\\
S_{\pi_1} = \pi_2,\pi_7,\pi_{10}: \quad &
\end{aligned}$$

Finding Maximal Intersectables (cont)

$$(\pi_1,\pi_2)(\pi_1,\pi_7)(\pi_1,\pi_{10})(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)$$
$$(\pi_4,\pi_8)(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\pi_7)(\pi_6,\overline{\pi_5})(\pi_7,\pi_{10})(\pi_8,\pi_9)(\pi_8,\overline{\pi_{10}})(\pi_9,\pi_{10})$$

$$\begin{aligned}
S_{\pi_3} = \pi_4,\pi_5,\pi_8,\pi_9: \quad & c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),\\
& (\pi_6,\overline{\pi_5}),(\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),\\
& (\pi_3,\pi_5)\}\\
S_{\pi_2} = \pi_6,\overline{\pi_5}: \quad & c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_7,\pi_{10}),(\pi_6,\pi_7),\\
& (\pi_4,\overline{\pi_5}),(\pi_3,\pi_4,\pi_8,\pi_9),(\pi_3,\pi_5),\\
& (\pi_2,\pi_6,\overline{\pi_5})\}\\
S_{\pi_1} = \pi_2,\pi_7,\pi_{10}: \quad & c = \{(\pi_9,\pi_{10}),(\pi_8,\overline{\pi_{10}}),(\pi_6,\pi_7),(\pi_4,\overline{\pi_5}),\\
& (\pi_3,\pi_4,\pi_8,\pi_9),(\pi_3,\pi_5),(\pi_2,\pi_6,\overline{\pi_5}),\\
& (\pi_1,\pi_7,\pi_{10}),(\pi_1,\pi_2)\}
\end{aligned}$$

Finding Maximal Intersectables (cont)

x_1	(π_1, π_2)
x_2	(π_1, π_7, π_{10})
x_3	$(\pi_2, \pi_6, \overline{\pi_5})$
x_4	$(\pi_3, \pi_4, \pi_8, \pi_9)$
x_5	(π_3, π_5)
x_6	$(\pi_4, \overline{\pi_5})$
x_7	(π_6, π_7)
x_8	$(\pi_8, \overline{\pi_{10}})$
x_9	(π_9, π_{10})

Setting up the Covering Problem

$$\mathbf{A} = \begin{array}{cccccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & \\
1 & 1 & - & - & - & - & - & - & - & \pi_1 \\
1 & - & 1 & - & - & - & - & - & - & \pi_2 \\
- & - & - & 1 & 1 & - & - & - & - & \pi_3 \\
- & - & - & 1 & - & 1 & - & - & - & \pi_4 \\
- & - & 1 & - & 1 & 1 & - & - & - & \pi_5 \\
- & - & 1 & - & - & - & 1 & - & - & \pi_6 \\
- & 1 & - & - & - & - & 1 & - & - & \pi_7 \\
- & - & - & 1 & - & - & - & 1 & - & \pi_8 \\
- & - & - & 1 & - & - & - & - & 1 & \pi_9 \\
- & 1 & - & - & - & - & - & 1 & 1 & \pi_{10}
\end{array}$$

Setting up the Covering Problem

Cyclic with $L = 3$

$$\mathbf{A} = \begin{array}{cccccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & \\
1 & 1 & - & - & - & - & - & - & - & \pi_1 \\
1 & - & 1 & - & - & - & - & - & - & \pi_2 \\
- & - & - & 1 & 1 & - & - & - & - & \pi_3 \\
- & - & - & 1 & - & 1 & - & - & - & \pi_4 \\
- & - & 1 & - & 1 & 1 & - & - & - & \pi_5 \\
- & - & 1 & - & - & - & 1 & - & - & \pi_6 \\
- & 1 & - & - & - & - & 1 & - & - & \pi_7 \\
- & - & - & 1 & - & - & - & 1 & - & \pi_8 \\
- & - & - & 1 & - & - & - & - & 1 & \pi_9 \\
- & 1 & - & - & - & - & - & 1 & 1 & \pi_{10}
\end{array}$$

Setting up the Covering Problem

Cyclic with $L = 3$

Branch on x_4

$$\mathbf{A} = \begin{array}{c} \\ \left[\begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \\ 1 & 1 & - & - & - & - & - & - & - \\ 1 & - & 1 & - & - & - & - & - & - \\ - & - & - & 1 & 1 & - & - & - & - \\ - & - & - & 1 & - & 1 & - & - & - \\ - & - & 1 & - & 1 & 1 & - & - & - \\ - & - & 1 & - & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 & - & - \\ - & - & - & 1 & - & - & - & 1 & - \\ - & - & - & 1 & - & - & - & - & 1 \\ - & 1 & - & - & - & - & - & 1 & 1 \end{array} \right] \end{array} \begin{array}{c} \\ \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \pi_6 \\ \pi_7 \\ \pi_8 \\ \pi_9 \\ \pi_{10} \end{array}$$

Reduced Covering Problem

$$\mathbf{A} = \begin{array}{c}\\ \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccccccc} & x_1 & x_2 & x_3 & x_5 & x_6 & x_7 & x_8 & x_9 & \\ \left[\begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\right. & 1 & 1 & - & - & - & - & - & - & \pi_1 \\ & 1 & - & 1 & - & - & - & - & - & \pi_2 \\ & - & - & 1 & 1 & 1 & - & - & - & \pi_5 \\ & - & - & 1 & - & - & 1 & - & - & \pi_6 \\ & - & 1 & - & - & - & 1 & - & - & \pi_7 \\ & - & 1 & - & - & - & - & 1 & 1 & \pi_{10} \end{array}$$

$$x_4 = 1$$

Reduced Covering Problem

Column x_3 dominates x_5 and x_6
Column x_2 dominates x_8 and x_9

$$\mathbf{A} = \begin{array}{cc} & \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} \\ & \left[\begin{array}{cccccccc} 1 & 1 & - & - & - & - & - & - \\ 1 & - & 1 & - & - & - & - & - \\ - & - & 1 & 1 & 1 & - & - & - \\ - & - & 1 & - & - & 1 & - & - \\ - & 1 & - & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 & 1 \end{array}\right] \begin{array}{c} \pi_1 \\ \pi_2 \\ \pi_5 \\ \pi_6 \\ \pi_7 \\ \pi_{10} \end{array} \end{array}$$

$$x_4 = 1$$

Reduced Covering Problem

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccc} & x_1 & x_2 & x_3 & x_7 & \\ & 1 & 1 & - & - & \pi_1 \\ & 1 & - & 1 & - & \pi_2 \\ & - & - & 1 & - & \pi_5 \\ & - & - & 1 & 1 & \pi_6 \\ & - & 1 & - & 1 & \pi_7 \\ & - & 1 & - & - & \pi_{10} \end{array}$$

$$x_4 = 1, x_5 = 0, x_6 = 0, x_8 = 0, x_9 = 0$$

Reduced Covering Problem

x_2 and x_3 are now essential

$$\mathbf{A} = \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_7 \end{array} \\ \left[\begin{array}{cccc} 1 & 1 & - & - \\ 1 & - & 1 & - \\ - & - & 1 & - \\ - & - & 1 & 1 \\ - & 1 & - & 1 \\ - & 1 & - & - \end{array}\right] \end{array} \begin{array}{c} \\ \pi_1 \\ \pi_2 \\ \pi_5 \\ \pi_6 \\ \pi_7 \\ \pi_{10} \end{array}$$

$$x_4 = 1, x_5 = 0, x_6 = 0, x_8 = 0, x_9 = 0$$

Minimal Boolean Matrix

$$\begin{array}{lcccccc} & a & b & c & d & e & f \\ x_2 : (\pi_1,\pi_7,\pi_{10}) & 0 & 0 & 1 & 0 & 1 & 1 \\ x_3 : (\pi_2,\overline{\pi_5},\pi_6) & 0 & 1 & 1 & 0 & 0 & 1 \\ x_4 : (\pi_3,\pi_4,\pi_8,\pi_9) & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$$

	$y_1 y_2 y_3$
a	000
b	011
c	110
d	001
e	101
f	111

Original Flow Table

	x_1	x_2	x_3	x_4
a	a,0	c,1	d,0	c,1
b	a,0	f,1	c,1	b,0
c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

Encoded Flow Table

	x_1	x_2	x_3	x_4
000	000,0	011,1	100,0	011,1
110	000,0	111,1	011,1	110,0
011	111,1	011,1	011,1	011,1
100	–,–	100,0	100,0	110,0
101	000,0	100,0	011,1	101,1
111	111,1	111,1	–,–	111,1

Fed-Back Outputs as State Variables

- Previously ignored outputs during state assignment.
- May be possible to feed back outputs as state variables.
- Determine in each state under each input the value of each output upon entry.
- This information can satisfy some partitions.
- Satisfying partitions, can reduce number of state variables.

Example Flow Table

	x_1	x_2	x_3	x_4
a	a,0	c,1	d,0	c,1
b	a,0	f,1	c,1	b,0
c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_1 &= \{ab; cf\} \\
\pi_2 &= \{ae; cf\} \\
\pi_3 &= \{ac; de\} \\
\pi_4 &= \{ac; bf\} \\
\pi_5 &= \{bf; de\} \\
\pi_6 &= \{ad; bc\} \\
\pi_7 &= \{ad; ce\} \\
\pi_8 &= \{ac; bd\} \\
\pi_9 &= \{ac; ef\} \\
\pi_{10} &= \{bd; ef\}
\end{aligned}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
b	a,0	f,1	c,1	b,0
c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_1 &= \{ab; cf\} \\
\pi_2 &= \{ae; cf\} \\
\pi_3 &= \{ac; de\} \\
\pi_4 &= \{ac; bf\} \\
\pi_5 &= \{bf; de\} \\
\pi_6 &= \{ad; bc\} \\
\pi_7 &= \{ad; ce\} \\
\pi_8 &= \{ac; bd\} \\
\pi_9 &= \{ac; ef\} \\
\pi_{10} &= \{bd; ef\}
\end{aligned}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_1 &= \{ab; cf\} \\
\pi_2 &= \{ae; cf\} \\
\pi_3 &= \{ac; de\} \\
\pi_4 &= \{ac; bf\} \\
\pi_5 &= \{bf; de\} \\
\pi_6 &= \{ad; bc\} \\
\pi_7 &= \{ad; ce\} \\
\pi_8 &= \{ac; bd\} \\
\pi_9 &= \{ac; ef\} \\
\pi_{10} &= \{bd; ef\}
\end{aligned}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
(1) c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_1 &= \{ab; cf\} \\
\pi_2 &= \{ae; cf\} \\
\pi_3 &= \{ac; de\} \\
\pi_4 &= \{ac; bf\} \\
\pi_5 &= \{bf; de\} \\
\pi_6 &= \{ad; bc\} \\
\pi_7 &= \{ad; ce\} \\
\pi_8 &= \{ac; bd\} \\
\pi_9 &= \{ac; ef\} \\
\pi_{10} &= \{bd; ef\}
\end{aligned}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
(1) c	f,1	c,1	c,1	c,1
(0) d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_1 &= \{ab;cf\} \\
\pi_2 &= \{ae;cf\} \\
\pi_3 &= \{ac;de\} \\
\pi_4 &= \{ac;bf\} \\
\pi_5 &= \{bf;de\} \\
\pi_6 &= \{ad;bc\} \\
\pi_7 &= \{ad;ce\} \\
\pi_8 &= \{ac;bd\} \\
\pi_9 &= \{ac;ef\} \\
\pi_{10} &= \{bd;ef\}
\end{aligned}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
(1) c	f,1	c,1	c,1	c,1
(0) d	–,–	d,0	d,0	b,0
(1) e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

$$\pi_1 = \{ab; cf\}$$
$$\pi_2 = \{ae; cf\}$$
$$\pi_3 = \{ac; de\}$$
$$\pi_4 = \{ac; bf\}$$
$$\pi_5 = \{bf; de\}$$
$$\pi_6 = \{ad; bc\}$$
$$\pi_7 = \{ad; ce\}$$
$$\pi_8 = \{ac; bd\}$$
$$\pi_9 = \{ac; ef\}$$
$$\pi_{10} = \{bd; ef\}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
(1) c	f,1	c,1	c,1	c,1
(0) d	–,–	d,0	d,0	b,0
(1) e	a,0	d,0	c,1	e,1
(1) f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_1 &= \{ab; cf\} \\
\pi_2 &= \{ae; cf\} \\
\pi_3 &= \{ac; de\} \\
\pi_4 &= \{ac; bf\} \\
\pi_5 &= \{bf; de\} \\
\pi_6 &= \{ad; bc\} \\
\pi_7 &= \{ad; ce\} \\
\pi_8 &= \{ac; bd\} \\
\pi_9 &= \{ac; ef\} \\
\pi_{10} &= \{bd; ef\}
\end{aligned}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
(1) c	f,1	c,1	c,1	c,1
(0) d	–,–	d,0	d,0	b,0
(1) e	a,0	d,0	c,1	e,1
(1) f	f,1	f,1	–,–	e,1

$$\pi_2 = \{ae; cf\}$$
$$\pi_3 = \{ac; de\}$$
$$\pi_4 = \{ac; bf\}$$
$$\pi_5 = \{bf; de\}$$
$$\pi_6 = \{ad; bc\}$$
$$\pi_7 = \{ad; ce\}$$
$$\pi_8 = \{ac; bd\}$$
$$\pi_9 = \{ac; ef\}$$
$$\pi_{10} = \{bd; ef\}$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
(1) c	f,1	c,1	c,1	c,1
(0) d	–,–	d,0	d,0	b,0
(1) e	a,0	d,0	c,1	e,1
(1) f	f,1	f,1	–,–	e,1

$$
\begin{aligned}
\pi_2 &= \{ae; cf\} \\
\pi_3 &= \{ac; de\} \\
\pi_4 &= \{ac; bf\} \\
\pi_5 &= \{bf; de\} \\
\pi_6 &= \{ad; bc\} \\
\pi_8 &= \{ac; bd\} \\
\pi_9 &= \{ac; ef\} \\
\pi_{10} &= \{bd; ef\}
\end{aligned}
$$

Example Flow Table

	x_1	x_2	x_3	x_4
(0) a	a,0	c,1	d,0	c,1
(0) b	a,0	f,1	c,1	b,0
(1) c	f,1	c,1	c,1	c,1
(0) d	–,–	d,0	d,0	b,0
(1) e	a,0	d,0	c,1	e,1
(1) f	f,1	f,1	–,–	e,1

$$\begin{aligned}
\pi_2 &= \{ae; cf\} \\
\pi_3 &= \{ac; de\} \\
\pi_4 &= \{ac; bf\} \\
\pi_5 &= \{bf; de\} \\
\pi_6 &= \{ad; bc\} \\
\pi_8 &= \{ac; bd\} \\
\pi_9 &= \{ac; ef\}
\end{aligned}$$

Modified Partition List

π_2 = { ae, cf }
π_3 = { ac, de }
π_4 = { ac, bf }
π_5 = { bf, de }
π_6 = { ad, bc }
π_8 = { ac, bd }
π_9 = { ac, ef }

	a	b	c	d	e	f
π_2	0	–	1	–	0	1
π_3	0	–	0	1	1	–
π_4	0	1	0	–	–	1
π_5	–	0	–	1	1	0
π_6	0	1	1	0	–	–
π_8	0	1	0	1	–	–
π_9	0	–	0	–	1	1

Pairwise Intersectibles

π_3	$\times$						
π_4	$\times$	$\sim$					
π_5	$\times$	$\sim$	$\times$				
π_6	$\sim$	$\times$	$\times$	$\times$			
π_8	$\times$	$\sim$	$\sim$	$\times$	$\times$		
π_9	$\times$	$\sim$	$\sim$	$\times$	$\times$	$\sim$	
$\overline{\pi_5}$	$\sim$	$\times$	$\sim$	$\times$	$\sim$	$\times$	$\times$
	π_2	π_3	π_4	π_5	π_6	π_8	π_9

$$(\pi_2,\pi_6)(\pi_2,\overline{\pi_5})(\pi_3,\pi_4)(\pi_3,\pi_5)(\pi_3,\pi_8)(\pi_3,\pi_9)(\pi_4,\pi_8)$$
$$(\pi_4,\pi_9)(\pi_4,\overline{\pi_5})(\pi_6,\overline{\pi_5})(\pi_8,\pi_9)$$

Maximal Intersectibles

First step: $c = \{(\pi_8, \pi_9)\}$

$S_{\pi_6} = \overline{\pi_5}$: $c = \{(\pi_8, \pi_9), (\pi_6, \overline{\pi_5}), \}$

$S_{\pi_4} = \pi_8, \pi_9, \overline{\pi_5}$: $c = \{(\pi_4, \pi_8, \pi_9), (\pi_4, \overline{\pi_5}), (\pi_6, \overline{\pi_5})\}$

$S_{\pi_3} = \pi_4, \pi_5, \pi_8, \pi_9$: $c = \{(\pi_3, \pi_4, \pi_8, \pi_9), (\pi_3, \pi_5), (\pi_4, \overline{\pi_5}), (\pi_6, \overline{\pi_5})\}$

$S_{\pi_2} = \pi_6, \overline{\pi_5}$: $c = \{(\pi_3, \pi_4, \pi_8, \pi_9), (\pi_3, \pi_5), (\pi_4, \overline{\pi_5}),$
$(\pi_2, \pi_6, \overline{\pi_5})\}$

Maximal Intersectables (cont)

x_1	$(\pi_2, \pi_6, \overline{\pi_5}$
x_2	$(\pi_3, \pi_4, \pi_8, \pi_9)$
x_3	(π_3, π_5)
x_4	$(\pi_4, \overline{\pi_5})$

Constraint Matrix

$$\mathbf{A} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & - & - & - & \pi_2 \\ - & 1 & 1 & - & \pi_3 \\ - & 1 & - & 1 & \pi_4 \\ 1 & - & 1 & 1 & \pi_5 \\ 1 & - & - & - & \pi_6 \\ - & 1 & - & - & \pi_8 \\ - & 1 & - & - & \pi_9 \end{array}$$

Constraint Matrix

x_1 and x_2 are essential.

$$\mathbf{A} = \begin{array}{c} \\ \\ \end{array} \begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & - & - & - & \pi_2 \\ - & 1 & 1 & - & \pi_3 \\ - & 1 & - & 1 & \pi_4 \\ 1 & - & 1 & 1 & \pi_5 \\ 1 & - & - & - & \pi_6 \\ - & 1 & - & - & \pi_8 \\ - & 1 & - & - & \pi_9 \end{array}$$

Minimal Boolean Matrix

$$\begin{array}{lcccccc}
 & a & b & c & d & e & f \\
x_1 : (\pi_2, \overline{\pi_5}, \pi_6) & 0 & 1 & 1 & 0 & 0 & 1 \\
x_2 : (\pi_3, \pi_4, \pi_8, \pi_9) & 0 & 1 & 0 & 1 & 1 & 1
\end{array}$$

	$y_1 y_2 y_3$
a	00
b	11
c	01
d	10
e	10
f	11

Original Flow Table

	x_1	x_2	x_3	x_4
a	a,0	c,1	d,0	c,1
b	a,0	f,1	c,1	b,0
c	f,1	c,1	c,1	c,1
d	–,–	d,0	d,0	b,0
e	a,0	d,0	c,1	e,1
f	f,1	f,1	–,–	e,1

New Encoded Flow Table

	x_1	x_2	x_3	x_4
00	00,0	01,1	10,0	01,1
11	00,0	11,1	01,1	11,0
01	11,1	01,1	01,1	01,1
10	–,–	10,0	10,0	11,0
10	00,0	10,0	01,1	10,1
11	11,1	11,1	–,–	11,1

Hazard-free Logic Synthesis

- For each next state and output signal:
  - Derive *sum-of-products* (SOP) implementation.
  - Transform SOP using laws of Boolean algebra into a multi-level logic implementation.
  - Map to gates found in the given gate library.
- For asynchronous FSMs, must avoid *hazards* in SOP.
- Some laws of Boolean algebra introduce hazards.
- First describe for SIC fundamental-mode.

Boolean Functions and Minterms

- A *Boolean function* f of n variables $x_1, x_2, \ldots, x_n$ is a mapping: $f : \{0,1\}^n \rightarrow \{0,1,-\}$.
- Each element m of $\{0,1\}^n$ is called a *minterm*.
- The value of a variable x_i in a minterm m is given by $m(i)$.
- The *ON-set* of f is the set of minterms which return 1.
- The *OFF-set* of f is the set of minterms which return 0.
- The *DC-set* of f is the set of minterms which return $-$.

Literals and Products

- A *literal* is either the variable, x_i, or its complement, x_i'.
- The literal x_i evaluates to 1 in the minterm m when $m(i) = 1$.
- The literal x_i' evaluates to 1 when $m(i) = 0$.
- A *product* is a conjunction (AND) of literals.
- A product evaluates to 1 for m (i.e., the product *contains* m) if each literal evaluates to 1 in m.
- $X \subseteq Y$ if minterms contained in X are a subset of those in Y.
- *Intersection* of two products is the minterms contained in both.
- A *sum-of-products* (SOP) is a set of products.
- A SOP contains m when a product in the SOP contains m.

Implicants and Prime Implicants

- An *implicant* is a product that contains none of the OFF-set.
- A *prime implicant* is an implicant contained by no other.
- A *cover* is a SOP which contains the entire ON-set and none of the OFF-set.
- A cover may optionally include part of the DC-set.
- The two-level logic minimization problem is to find a minimum-cost cover of the function.
- For SIC fundamental-mode, a minimal cover is always composed of only prime implicants.

Two-Level Logic Minimization Example

			wx		
		00	01	11	10
	00	1	1	1	1
yz	01	0	1	1	—
	11	0	1	1	0
	10	0	—	0	0

Two-Level Logic Minimization Example

		wx			
		00	01	11	10
	00	1	1	1	1
yz	01	0	1	1	–
	11	0	1	1	0
	10	0	–	0	0

$$\text{ON-set} \quad = \quad \{\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}\,\overline{z}, wx\overline{y}\,\overline{z}, w\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}z, wx\overline{y}z, \overline{w}xyz, wxyz\}$$

Two-Level Logic Minimization Example

		wx			
		00	01	11	10
	00	1	1	1	1
yz	01	0	1	1	–
	11	0	1	1	0
	10	0	–	0	0

$$\begin{aligned}\text{ON-set} &= \{\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}\,\overline{z}, wx\overline{y}\,\overline{z}, w\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}z, wx\overline{y}z, \overline{w}xyz, wxyz\}\\ \text{OFF-set} &= \{\overline{w}\,\overline{x}\,\overline{y}z, \overline{w}\,\overline{x}yz, w\overline{x}yz, \overline{w}\,\overline{x}y\overline{z}, wxy\overline{z}, w\overline{x}y\overline{z}\}\end{aligned}$$

Two-Level Logic Minimization Example

		wx			
		00	01	11	10
yz	00	1	1	1	1
	01	0	1	1	–
	11	0	1	1	0
	10	0	–	0	0

$$\begin{aligned}
\text{ON-set} &= \{\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}\,\overline{z}, wx\overline{y}\,\overline{z}, w\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}z, wx\overline{y}z, \overline{w}xyz, wxyz\} \\
\text{OFF-set} &= \{\overline{w}\,\overline{x}\,\overline{y}z, \overline{w}\,\overline{x}yz, w\overline{x}yz, \overline{w}\,\overline{x}y\overline{z}, wxy\overline{z}, w\overline{x}y\overline{z}\} \\
\text{DC-set} &= \{w\overline{x}\,\overline{y}z, \overline{w}xy\overline{z}\}
\end{aligned}$$

Prime Implicant Generation

- For functions of less than 4 variables, can use a Karnaugh map.
- For more variables, Karnaugh maps too tedious.
- Quine's tabular method is better but requires all minterms be listed.
- Recursive procedure based on *consensus* and *complete sums* is better.

Consensus and Complete Sums

- The consensus theorem states: $xy + \overline{x}z = xy + \overline{x}z + yz$.
- The product yz is called the consensus for xy and $\overline{x}z$.
- A complete sum is defined to be a SOP formula composed of all the prime implicants.
- **Theorem 5.5** (Blake, 1937) A SOP is a complete sum iff:
    1. No term includes any other term.
    2. The consensus of any two terms of the formula either does not exist or is contained in some term of the formula.

Recursive Prime Generation

- **Theorem 5.6** (Blake, 1937) If we have two complete sums f_1 and f_2, we can obtain the complete sum for $f_1 \cdot f_2$ using the following two steps:
  1. Multiply out f_1 and f_2 using the following properties
     - $x \cdot x = x$ (*idempotent*)
     - $x \cdot (y + z) = xy + xz$ (*distributive*)
     - $x \cdot \overline{x} = 0$ (*complement*)
  2. Eliminate all terms contained in some other term.
- A recursive procedure for finding the complete sum for f:

$$\begin{aligned} \mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \\ &\quad \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \end{aligned}$$

where $abs(f)$ removes absorbed terms from f ($abs(a + ab) = a$).

Two-Level Logic Minimization Example

wx

		00	01	11	10
	00	1	1	1	1
yz	01	0	1	1	–
	11	0	1	1	0
	10	0	–	0	0

$$\begin{aligned}
\text{ON-set} &= \{\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}\,\overline{z}, wx\overline{y}\,\overline{z}, w\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}z, wx\overline{y}z, \overline{w}xyz, wxyz\} \\
\text{OFF-set} &= \{\overline{w}\,\overline{x}\,\overline{y}z, \overline{w}\,\overline{x}yz, w\overline{x}yz, \overline{w}\,\overline{x}y\overline{z}, wxy\overline{z}, w\overline{x}y\overline{z}\} \\
\text{DC-set} &= \{w\overline{x}\,\overline{y}z, \overline{w}xy\overline{z}\}
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w, x, y, z) &= y'z' + xz + wx'y'z + w'xyz'
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w, x, y, z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w, x, y, 0) &=
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\text{cs}(f) &= \text{abs}([x_1 + \text{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \text{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w, x, y, z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w, x, y, 0) &= y' + w'xy
\end{aligned}$$

Recursive Prime Generation: Example

$$
\begin{aligned}
\text{cs}(f) &= \text{abs}([x_1 + \text{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \text{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w, x, y, z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w, x, y, 0) &= y' + w'xy \\
f(w, x, 0, 0) &=
\end{aligned}
$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &=
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\text{cs}(f) &= \text{abs}([x_1 + \text{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \text{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w, x, y, z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w, x, y, 0) &= y' + w'xy \\
f(w, x, 0, 0) &= 1 \\
f(w, x, 1, 0) &= w'x
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\text{cs}(f) &= \text{abs}([x_1 + \text{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \text{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w, x, y, z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w, x, y, 0) &= y' + w'xy \\
f(w, x, 0, 0) &= 1 \\
f(w, x, 1, 0) &= w'x \\
cs(f(w, x, y, 0)) &=
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w, x, y, z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w, x, y, 0) &= y' + w'xy \\
f(w, x, 0, 0) &= 1 \\
f(w, x, 1, 0) &= w'x \\
cs(f(w, x, y, 0)) &= abs((y + 1)(y' + w'x)) = y' + w'x \\
f(w, x, y, 1) &=
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y'
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &=
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy'
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &=
\end{aligned}$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &= 1
\end{aligned}$$

Recursive Prime Generation: Example

$$
\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &= 1 \\
cs(f(w,x,y,1)) &=
\end{aligned}
$$

Recursive Prime Generation: Example

$$
\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &= 1 \\
cs(f(w,x,y,1)) &= abs((x+wy')(x'+1)) = x + wy'
\end{aligned}
$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &= 1 \\
cs(f(w,x,y,1)) &= abs((x+wy')(x'+1)) = x + wy' \\
cs(f(w,x,y,z)) &=
\end{aligned}$$

Recursive Prime Generation: Example

$$
\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &= 1 \\
cs(f(w,x,y,1)) &= abs((x+wy')(x'+1)) = x + wy' \\
cs(f(w,x,y,z)) &= abs((z+y'+w'x)(z'+x+wy'))
\end{aligned}
$$

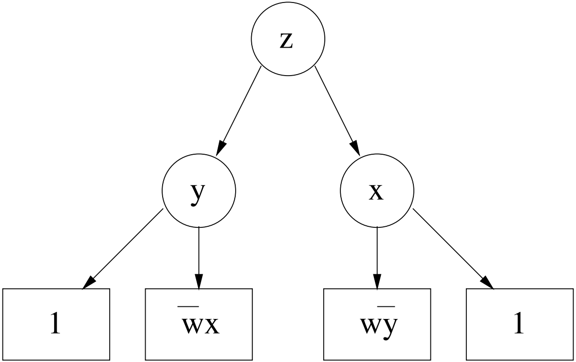
Recursive Prime Generation: Example

$$
\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &= 1 \\
cs(f(w,x,y,1)) &= abs((x+wy')(x'+1)) = x + wy' \\
cs(f(w,x,y,z)) &= abs((z+y'+w'x)(z'+x+wy')) \\
&= abs(xz + wy'z + y'z' + xy' + wy' + w'xz' + w'x)
\end{aligned}
$$

Recursive Prime Generation: Example

$$\begin{aligned}
\mathrm{cs}(f) &= \mathrm{abs}([x_1 + \mathrm{cs}(f(0, x_2, \ldots, x_n))] \cdot [\overline{x_1} + \mathrm{cs}(f(1, x_2, \ldots, x_n))]) \\
f(w,x,y,z) &= y'z' + xz + wx'y'z + w'xyz' \\
f(w,x,y,0) &= y' + w'xy \\
f(w,x,0,0) &= 1 \\
f(w,x,1,0) &= w'x \\
cs(f(w,x,y,0)) &= abs((y+1)(y'+w'x)) = y' + w'x \\
f(w,x,y,1) &= x + wx'y' \\
f(w,0,y,1) &= wy' \\
f(w,1,y,1) &= 1 \\
cs(f(w,x,y,1)) &= abs((x+wy')(x'+1)) = x + wy' \\
cs(f(w,x,y,z)) &= abs((z+y'+w'x)(z'+x+wy')) \\
&= abs(xz + wy'z + y'z' + xy' + wy' + w'xz' + w'x) \\
&= xz + y'z' + xy' + wy' + w'x
\end{aligned}$$

Recursion Tree for Example



Prime Implicant Selection

	xz	$\overline{y}\,\overline{z}$	$x\overline{y}$	$w\overline{y}$	$\overline{w}x$
$\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}$	—	1	—	—	—
$\overline{w}x\overline{y}\,\overline{z}$	—	1	1	—	1
$wx\overline{y}\,\overline{z}$	—	1	1	1	—
$w\overline{x}\,\overline{y}\,\overline{z}$	—	1	—	1	—
$\overline{w}x\overline{y}z$	1	—	1	—	1
$wx\overline{y}z$	1	—	1	1	—
$\overline{w}xyz$	1	—	—	—	1
$wxyz$	1	—	—	—	—

Prime Implicant Selection

	xz	$\overline{y}\,\overline{z}$	$x\overline{y}$	$w\overline{y}$	$\overline{w}x$
$\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}$	–	1	–	–	–
$\overline{w}x\overline{y}\,\overline{z}$	–	1	1	–	1
$wx\overline{y}\,\overline{z}$	–	1	1	1	–
$w\overline{x}\,\overline{y}\,\overline{z}$	–	1	–	1	–
$\overline{w}x\overline{y}z$	1	–	1	–	1
$wx\overline{y}z$	1	–	1	1	–
$\overline{w}xyz$	1	–	–	–	1
$wxyz$	1	–	–	–	–

Solution: $f = xz + \overline{y}\,\overline{z}$

Combinational Hazards

- For asynchronous design, two-level logic minimization problem is complicated by hazards.
- Let us consider the design of a function f to implement either an output or next state variable.
- When input changes under SIC, circuit moves from minterm m_1 to another m_2 which differ in value in exactly one x_i.
- During this transition, there are four possible transitions of f:
  1. *Static* $0 \rightarrow 0$ *transition*: $f(m_1) = f(m_2) = 0$.
  2. *Static* $1 \rightarrow 1$ *transition*: $f(m_1) = f(m_2) = 1$.
  3. *Dynamic* $0 \rightarrow 1$ *transition*: $f(m_1) = 0$ and $f(m_2) = 1$.
  4. *Dynamic* $1 \rightarrow 0$ *transition*: $f(m_1) = 1$ and $f(m_2) = 0$.

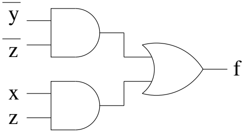
Static 0-Hazard

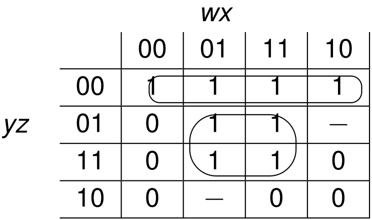
- If during a static 0 → 0 transition, the cover of f can due to differences in delays momentarily evaluate to 1, then we say that there exists a *static 0-hazard*.
- In a SOP cover of a function, no product term is allowed to include either m_1 or m_2 since they are in the OFF-set.
- Static 0-hazard exists only if some product includes both x_i and $\overline{x_i}$.
- Such a product is not useful since it contains no minterms.
- If we exclude such product terms from the cover, then the SOP cover can never produce a static 0-hazard.

Static 1-Hazard

- If during a static 1 → 1 transition, the cover of f can evaluate to 0, then we say that there exists a *static 1-hazard*.
- Consider case where one product p_1 contains m_1 but not m_2 and another product p_2 contains m_2 but not m_1.
- If p_1 is implemented with a faster gate than p_2, then the gate for p_1 can turn off faster than the gate for p_2 turns on which can lead to the cover momentarily evaluating to a 0.
- To eliminate static 1-hazards, for each $m_1 \longrightarrow m_2$, there must exist a product in the cover that includes both m_1 and m_2.

Static 1-Hazard Example





		wx			
		00	01	11	10
yz	00	1	1	1	1
	01	0	1	1	–
	11	0	1	1	0
	10	0	–	0	0

Dynamic Hazards

- If during a 0 $\longrightarrow$ 1 transition, the cover can change from 0 to 1 back to 0 and finally stabilize at 1, we say the cover has a *dynamic* 0 $\longrightarrow$ 1 *hazard*.
- Assuming no useless product terms (ones that include both x_i and $\overline{x_i}$), this is impossible under the SIC assumption.
- No product is allowed to include m_1 since it is in the OFF-set.
- Any product that includes m_2 turns on monotonically.
- Similarly, there are no *dynamic* 1 $\longrightarrow$ 0 *hazards*.

Removing Hazards

- A simple, inefficent approach to produce a hazard-free SOP cover is to include all prime implicants in the cover.
- Since two minterms m_1 and m_2 in a transition are distance 1 apart, they must be included together in some prime.
- An implicant exists which is made up of all literals that are equal in both m_1 and m_2.
- This implicant must be part of some prime implicant.
- For our example, the following cover is guaranteed to be hazard-free under SIC:

$$f = xz + \overline{y}\,\overline{z} + x\overline{y} + w\overline{y} + \overline{w}x$$

Better Approach to Remove Hazards

- Form an implicant out of each pair of states m_1 and m_2 involved in a static $1 \longrightarrow 1$ transition which includes each literal that is the same value in both m_1 and m_2.
- The covering problem is now to find the minimum number of prime implicants that cover each of these *transition cubes*.

Two-Level Logic Minimization Example

		wx			
		00	01	11	10
	00	1	1	1	1
yz	01	0	1	1	–
	11	0	1	1	0
	10	0	–	0	0

$$
\begin{array}{rcl}
\text{ON-set} & = & \{\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}\,\overline{z}, wx\overline{y}\,\overline{z}, w\overline{x}\,\overline{y}\,\overline{z}, \overline{w}x\overline{y}z, wx\overline{y}z, \overline{w}xyz, wxyz\} \\
\text{OFF-set} & = & \{\overline{w}\,\overline{x}\,\overline{y}z, \overline{w}\,\overline{x}yz, w\overline{x}yz, \overline{w}\,\overline{x}y\overline{z}, wxy\overline{z}, w\overline{x}y\overline{z}\} \\
\text{DC-set} & = & \{w\overline{x}\,\overline{y}z, \overline{w}xy\overline{z}\}
\end{array}
$$

2-Level Hazard-Free Synthesis: Example

	xz	$\bar{y}\bar{z}$	$x\bar{y}$	$w\bar{y}$	$\bar{w}x$
$\bar{w}\bar{y}\bar{z}$	–	1	–	–	–
$\bar{x}\bar{y}\bar{z}$	–	1	–	–	–
$\bar{w}x\bar{y}$	–	–	1	–	1
$wx\bar{y}$	–	–	1	1	–
$w\bar{y}\bar{z}$	–	1	–	1	–
$x\bar{y}z$	1	–	1	–	–
$\bar{w}xz$	1	–	–	–	1
wxz	1	–	–	–	–

2-Level Hazard-Free Synthesis: Example

	xz	$\overline{y}\,\overline{z}$	$x\overline{y}$	$w\overline{y}$	$\overline{w}x$
$\overline{w}\,\overline{y}\,\overline{z}$	–	1	–	–	–
$\overline{x}\,\overline{y}\,\overline{z}$	–	1	–	–	–
$\overline{w}x\overline{y}$	–	–	1	–	1
$wx\overline{y}$	–	–	1	1	–
$w\overline{y}\overline{z}$	–	1	–	1	–
$x\overline{y}z$	1	–	1	–	–
$\overline{w}xz$	1	–	–	–	1
wxz	1	–	–	–	–

Solution: $f = xz + \overline{y}\,\overline{z}$

2-Level Hazard-Free Synthesis: Example

	$x\overline{y}$	$w\overline{y}$	$\overline{w}x$
$\overline{w}x\overline{y}$	1	$-$	1
$wx\overline{y}$	1	1	$-$

Solution: $f = xz + \overline{y}\,\overline{z}$

2-Level Hazard-Free Synthesis: Example

	$x\overline{y}$
$\overline{w}x\overline{y}$	1
$wx\overline{y}$	1

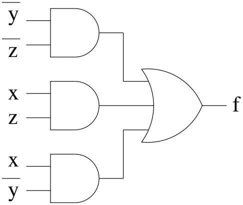
Solution: $f = xz + \overline{y}\,\overline{z}$

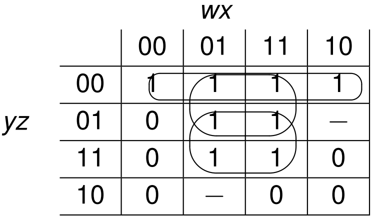
2-Level Hazard-Free Synthesis: Example

	$x\overline{y}$
$\overline{w}x\overline{y}$	1
$wx\overline{y}$	1

Solution: $f = xz + \overline{y}\,\overline{z} + x\overline{y}$

2-Level Hazard-Free Synthesis: Example





	00	01	11	10
00	1	1	1	1
01	0	1	1	–
11	0	1	1	0
10	0	–	0	0

Extensions for MIC Operation

- Preceeding restricted the class of circuits to SIC.
- Each input burst can have only a single transition.
- Now extend the synthesis method to MIC.
- Synthesize any XBM machine satisfying the maximal set property.

Transition Cubes

- MIC Transitions begin in one minterm m_1 and end in another m_2 where the values of multiple variables may have changed.
- m_1 is called the *start point* while m_2 is called the *end point*.
- Machine may pass through minterms between m_1 and m_2.
- Set of minterms is called a *transition cube* (denoted $[m_1, m_2]$).
- Transition cube can be represented with a product which contains a literal for each x_i in which $m_1(i) = m_2(i)$.
- *Open transition cube* $[m_1, m_2)$ includes all minterms in $[m_1, m_2]$ except m_2.
- An open transition cube represented using a set of products.

Transition Cube: Example

		wx			
		00	01	11	10
yz	00	1	1	1	1
	01	0	1	1	1
	11	0	1	1	0
	10	0	1	0	0

$$[\overline{w}\,x\,\overline{y}\,\overline{z},\; w\,x\,\overline{y}\,z]$$

$$[\overline{w}\,x\,y\,z,\; w\,x\,y\,\overline{z})$$

Function Hazards

- If f does not change monotonically during a multiple-input change, f has a *function hazard* for that transition.
- A function f contains a function hazard during a transition from m_1 to m_2 if there exists an m_3 and m_4 such that:
  1. $m_3 \neq m_1$ and $m_4 \neq m_2$.
  2. $m_3 \in [m_1, m_2]$ and $m_4 \in [m_3, m_2]$.
  3. $f(m_1) \neq f(m_3)$ and $f(m_4) \neq f(m_2)$.
- If $f(m_1) = f(m_2)$, it is a *static function hazard*.
- If $f(m_1) \neq f(m_2)$, it is a *dynamic function hazard*.

Function Hazards: Example

		wx			
		00	01	11	10
	00	1	1	1	1
yz	01	0	1	1	1
	11	0	1	1	0
	10	0	1	0	0

$$[\overline{w}\,\overline{x}\,\overline{y}\,\overline{z}, \overline{w}\,x\,\overline{y}\,z]$$
$$[\overline{w}\,x\,y\,\overline{z}, w\,\overline{x}\,y\,z]$$

Function Hazards

- If a transition has a function hazard, there is no implementation of the function which avoids the hazard during the transition.
- Fortunately, the synthesis method never produces a design with a transition that has a function hazard.

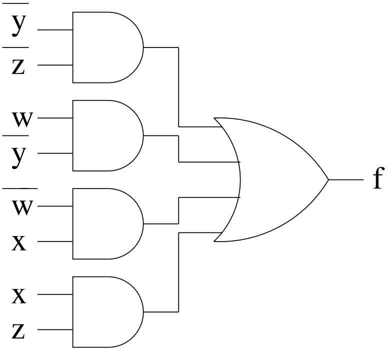
Combinational Hazards for State Variables

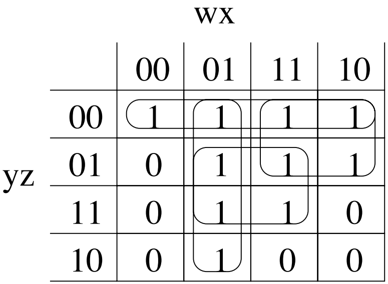
- A minimum transition time state assignment has MIC hazards.
- Multiple changing next state variables may be fed back to the input of the FSM.
- The circuit moves from one minterm m_1 to another minterm m_2, but multiple state variables may be changing concurrently.
- For normal flow tables with outputs that change only in unstable states then only static transitions possible.

MIC Static Hazards

- There can be no static 0-hazards.
- Since multiple variables may be changing concurrently, the cover may pass through other minterms between m_1 and m_2.
- To be free of static 1-hazards, it is necessary that a single product in the cover include all these minterms.
- Each $[m_1, m_2]$ where $f(m_1) = f(m_2) = 1$, must be contained in some product in the cover to eliminate static 1-hazards.

MIC Static Hazards: Example



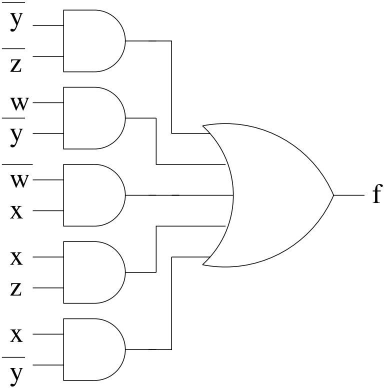


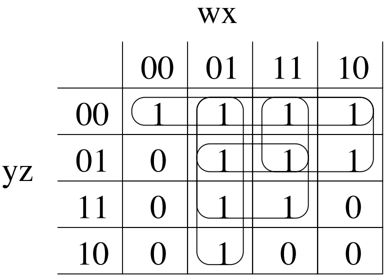
wx

yz	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	0	1	1	0
10	0	1	0	0

$[\overline{w}\,x\,\overline{y}\,\overline{z},\ w\,x\,\overline{y}\,z]$

MIC Static Hazards: Example





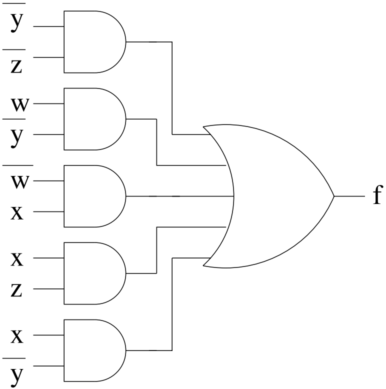
yz \ wx	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	0	1	1	0
10	0	1	0	0

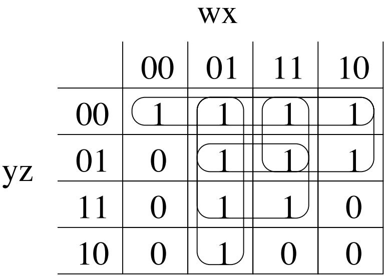
$[\overline{w}\,x\,\overline{y}\,\overline{z}, w\,x\,\overline{y}\,z]$

MIC Dynamic Hazards

- For each 1 → 0 transition, $[m_1, m_2]$, if a product in the cover intersects $[m_1, m_2]$, then it must include the start point, m_1.
- For each 0 → 1 transition, $[m_1, m_2]$, if a product in the cover intersects $[m_1, m_2]$, then it must include the end point, m_2.

MIC Dynamic Hazards: Example

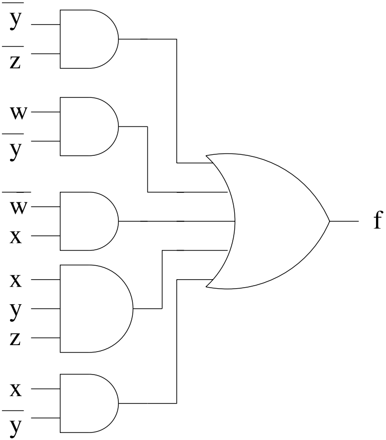


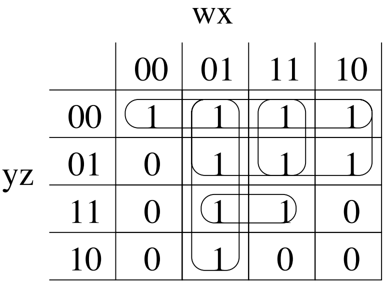


yz \ wx	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	0	1	1	0
10	0	1	0	0

$$[\overline{w}\,x\,\overline{y}\,\overline{z}, \overline{w}\,\overline{x}\,\overline{y}\,z]$$

MIC Dynamic Hazards: Example

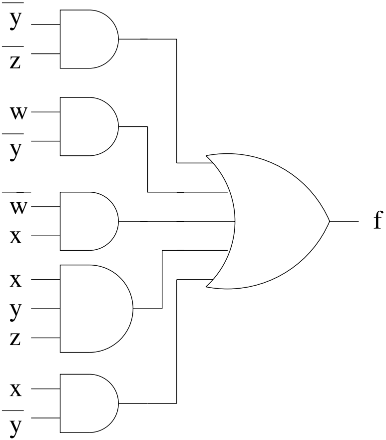


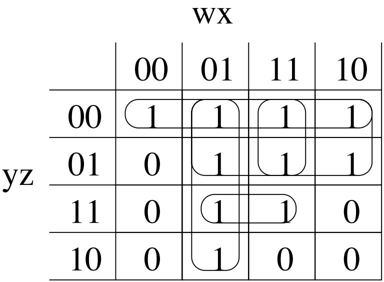


	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	0	1	1	0
10	0	1	0	0

$$[\overline{w}\, x\, \overline{y}\, \overline{z}, \overline{w}\, \overline{x}\, \overline{y}\, z]$$

MIC Dynamic Hazards: Example

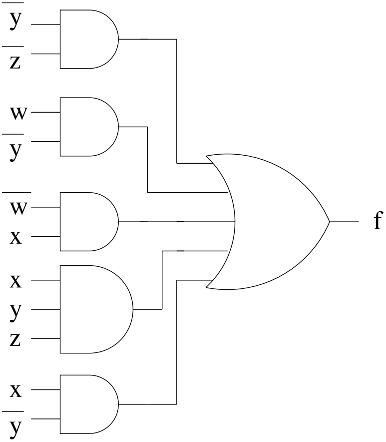


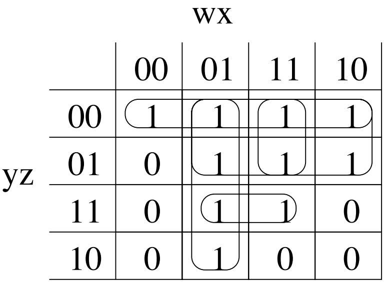


yz \ wx	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	0	1	1	0
10	0	1	0	0

$$[\overline{w}\,x\,\overline{y}\,\overline{z}, \overline{w}\,\overline{x}\,\overline{y}\,z]$$

MIC Dynamic Hazards: Example





yz \ wx	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	0	1	1	0
10	0	1	0	0

$$[\overline{w}\,x\,\overline{y}\,z, wxyz]$$

Burst-Mode Transitions

- In legal BM machines, types of transitions are restricted.
- A function may only change value after every transition in the input burst has occurred.
- $[m_1, m_2]$ for a function f is a *burst-mode transition* if for every minterm $m_i \in [m_1, m_2)$, $f(m_1) = f(m_i)$.
- The result is that if a function f only has burst-mode transitions, then it is free of function hazards.
- Also, any dynamic $0 \rightarrow 1$ transition is free of dynamic hazards.
- For any legal BM machine, there exists a hazard-free cover for each output and next state variable before state minimization.

Example Burst-Mode Transitions

y

x

	0	1
0	0	0
1	0	0

x

	0	1
0	1	1
1	1	1

x

	0	1
0	0	0
1	0	1

x

	0	1
0	1	1
1	1	0

y

x

	0	1
0	0	1
1	0	0

x

	0	1
0	1	0
1	1	1

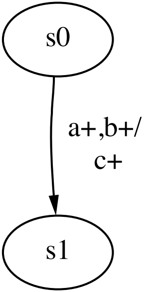
x

	0	1
0	0	1
1	1	1

x

	0	1
0	1	0
1	0	0

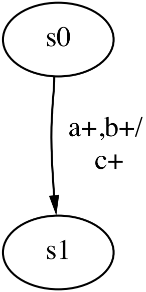
Burst-Mode Machine to Flow Table



	00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s0,0	01
s1			s1,1		10

(column label: *ab*)

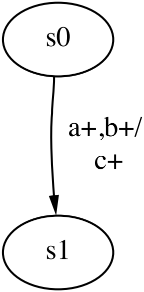
Burst-Mode Machine to Flow Table



	ab				
	00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s0,0	01
s1			s1,1		10

$$[\overline{a}\,\overline{b}\,\overline{x}y, ab\overline{x}y]$$

Burst-Mode Machine to Flow Table

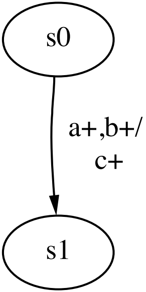


	ab				
	00	01	11	10	xy
s0	s0,0	s0,0	s1,1	s0,0	01
s1			s1,1		10

$$[\bar{a}\bar{b}\bar{x}y, ab\bar{x}y]$$

Dynamic 0 → 1 transition for output c and next-state variable X

Burst-Mode Machine to Flow Table



	00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s0,0	01
s1			s1,1		10

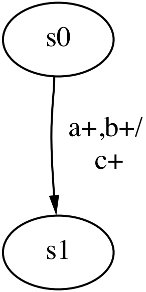
(The column labels 00, 01, 11, 10 are values of *ab*.)

$$[\bar{a}\,\bar{b}\,\bar{x}y, ab\bar{x}y]$$

Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Dynamic 1 → 0 transition for next-state variable *Y*

Burst-Mode Machine to Flow Table



	ab				
	00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s0,0	01
s1			s1,1		10

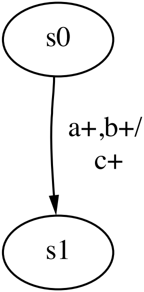
$$[\overline{a}\,\overline{b}\,\overline{x}y, ab\overline{x}y]$$

Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Dynamic 1 → 0 transition for next-state variable *Y*

$$[ab\overline{x}y, abx\overline{y}]$$

Burst-Mode Machine to Flow Table



	ab				
	00	01	11	10	xy
s0	s0,0	s0,0	s1,1	s0,0	01
s1			s1,1		10

$$[\bar{a}\bar{b}\bar{x}y, ab\bar{x}y]$$

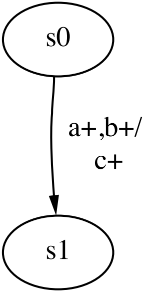
Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Dynamic 1 → 0 transition for next-state variable *Y*

$$[ab\bar{x}y, abx\bar{y}]$$

Static 1 → 1 transition for output *c* and next-state variable *X*

Burst-Mode Machine to Flow Table



	ab				
	00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s0,0	01
s1			s1,1		10

$$[\bar{a}\bar{b}\bar{x}y, ab\bar{x}y]$$

Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Dynamic 1 → 0 transition for next-state variable *Y*

$$[ab\bar{x}y, abx\bar{y}]$$

Static 1 → 1 transition for output *c* and next-state variable *X*

Static 0 → 0 transition for next-state variable *Y*

State Minimization: Burst-Mode

- After state minimization, it is possible that no hazard-free cover exists for some variable in the design.

Inputs a b c

State

	000	001	011	010	110	111	101	100
A	A,1	C,0	–	A,1	B,0	–	–	A,1

…

D	–	–	–	–	–	–	E,1	D,1

Reduce to:

AD	A,1	C,0	–	A,1	B,0	–	E,1	A,1

Static 1 → 1 transition $[a\overline{b}\,\overline{c}, a\overline{b}c]$

Dynamic 1 → 0 transition $[\overline{a}\,\overline{b}\,\overline{c}, ab\overline{c}]$

DHF-Compatibles

- Two states s_1 and s_2 are *dhf-compatible* when they are compatible and for each output z and transition $[m_1, m_2]$ of s_1 and for each transition $[m_3, m_4]$ of s_2:
  1. If z has a $1 \longrightarrow 0$ transition in $[m_1, m_2]$ and a $1 \longrightarrow 1$ transition in $[m_3, m_4]$, then $[m_1, m_2] \cap [m_3, m_4] = \emptyset$ or $m_1' \in [m_3, m_4]$.
  2. If z has a $1 \longrightarrow 0$ transition in $[m_1, m_2]$ and a $1 \longrightarrow 0$ transition in $[m_3, m_4]$, then $[m_1, m_2] \cap [m_3, m_4] = \emptyset$, $m_1 = m_3$, $[m_1, m_2] \subseteq [m_3, m_4]$, or $[m_3, m_4] \subseteq [m_1, m_2]$.

Required Cubes

- Transition cubes for each $1 \longrightarrow 1$ transition are *required cubes*.
- The end point of the transition cube for a $0 \longrightarrow 1$ transition is a required cube.
- *Transition subcubes* for each $1 \longrightarrow 0$ transition are required cubes.
- The transition subcubes for $1 \longrightarrow 0$ transition $[m_1, m_2]$ are all cubes of the form $[m_1, m_3]$ such that $f(m_3) = 1$.
- Can eliminate any subcube contained in another.
- The union of the required cubes forms the ON-set.
- Each of the required cubes must be contained in some product of the cover to insure hazard-freedom.

Required Cubes: BM Example

		ab			
		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}]$$

$$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$$

$$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$$

$$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$$

Required Cubes: BM Example

		ab			
		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\overline{b}\overline{c}d, ab\overline{c}\overline{d}] \quad 1 \longrightarrow 1$$

$$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$$

$$t_3 = [\overline{a}b\overline{c}\overline{d}, \overline{a}\overline{b}\overline{c}d]$$

$$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$$

Required Cubes: BM Example

ab

		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}] \quad 1 \longrightarrow 1 \quad a\overline{c}$$

$$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$$

$$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$$

$$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$$

Required Cubes: BM Example

ab

		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}] \quad 1 \longrightarrow 1 \quad a\overline{c}$$

$$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd] \quad 0 \longrightarrow 0$$

$$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$$

$$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$$

Required Cubes: BM Example

		ab			
		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$t_1 = [a\,\overline{b}\,\overline{c}\,d, a\,b\,\overline{c}\,\overline{d}]$ $1 \longrightarrow 1$ $a\overline{c}$

$t_2 = [a\,\overline{b}\,c\,\overline{d}, a\,\overline{b}\,c\,d]$ $0 \longrightarrow 0$ no required cubes

$t_3 = [\overline{a}\,b\,\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}\,d]$

$t_4 = [\overline{a}\,b\,c\,d, a\,\overline{b}\,c\,\overline{d}]$

Required Cubes: BM Example

		ab			
		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}]$ $\quad 1 \longrightarrow 1 \quad a\overline{c}$

$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$ $\quad 0 \longrightarrow 0$ no required cubes

$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$ $\quad 1 \longrightarrow 0$

$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$

Required Cubes: BM Example

ab

		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\overline{b}\,\overline{c}\,d, a\,b\,\overline{c}\,\overline{d}] \quad 1 \longrightarrow 1 \quad a\overline{c}$$

$$t_2 = [a\overline{b}\,c\,\overline{d}, a\overline{b}\,c\,d] \quad 0 \longrightarrow 0 \quad \text{no required cubes}$$

$$t_3 = [\overline{a}\,b\,\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}\,d] \quad 1 \longrightarrow 0 \quad \overline{a}\,\overline{c}\,\overline{d}, \overline{a}b\overline{c}$$

$$t_4 = [\overline{a}\,b\,c\,d, a\overline{b}\,c\,\overline{d}]$$

Required Cubes: BM Example

		ab			
		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}] \quad 1 \longrightarrow 1 \quad a\overline{c}$$

$$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd] \quad 0 \longrightarrow 0 \quad \text{no required cubes}$$

$$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d] \quad 1 \longrightarrow 0 \quad \overline{a}\,\overline{c}\,\overline{d}, \overline{a}b\overline{c}$$

$$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}] \quad 1 \longrightarrow 0$$

Required Cubes: BM Example

		ab			
		00	01	11	10
cd	00	1	1	1	1
	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$t_1 = [a\overline{b}\,\overline{c}\,d, a\,b\,\overline{c}\,\overline{d}]$ $1 \longrightarrow 1$ $a\overline{c}$

$t_2 = [a\overline{b}\,c\,\overline{d}, a\overline{b}\,c\,d]$ $0 \longrightarrow 0$ no required cubes

$t_3 = [\overline{a}\,b\,\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}\,d]$ $1 \longrightarrow 0$ $\overline{a}\,\overline{c}\,\overline{d}, \overline{a}b\overline{c}$

$t_4 = [\overline{a}\,b\,c\,d, a\overline{b}\,c\,\overline{d}]$ $1 \longrightarrow 0$ $bcd, \overline{a}c$

Required Cubes: BM Example

ab

		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\bar{b}\bar{c}d, ab\bar{c}\bar{d}] \quad 1 \longrightarrow 1 \quad a\bar{c}$$

$$t_2 = [a\bar{b}c\bar{d}, a\bar{b}cd] \quad 0 \longrightarrow 0 \quad \text{no required cubes}$$

$$t_3 = [\bar{a}b\bar{c}\bar{d}, \bar{a}\bar{b}\bar{c}d] \quad 1 \longrightarrow 0 \quad \bar{a}\bar{c}\bar{d}, \bar{a}b\bar{c}$$

$$t_4 = [\bar{a}bcd, a\bar{b}c\bar{d}] \quad 1 \longrightarrow 0 \quad bcd, \bar{a}c$$

$$\text{req-set} \quad = \quad \{a\bar{c}, \bar{a}\bar{c}\bar{d}, \bar{a}b\bar{c}, bcd, \bar{a}c\}$$

Prime Implicants: Example

$$f(a,b,c,d) \quad = \quad a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c$$

Prime Implicants: Example

$$
\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &=
\end{aligned}
$$

Prime Implicants: Example

$$
\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b
\end{aligned}
$$

Prime Implicants: Example

$$\begin{aligned} f(a,b,c,d) &= a\overline{c}+\overline{a}\,\overline{c}\,\overline{d}+\overline{a}b\overline{c}+bcd+\overline{a}c \\ f(a,b,0,d) &= a+\overline{a}\,\overline{d}+\overline{a}b \\ f(0,b,0,d) &= \end{aligned}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c}+\overline{a}\,\overline{c}\,\overline{d}+\overline{a}b\overline{c}+bcd+\overline{a}c \\
f(a,b,0,d) &= a+\overline{a}\,\overline{d}+\overline{a}b \\
f(0,b,0,d) &= \overline{d}+b
\end{aligned}$$

Prime Implicants: Example

$$
\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &=
\end{aligned}
$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1
\end{aligned}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &=
\end{aligned}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b
\end{aligned}$$

Prime Implicants: Example

$$\begin{array}{rcl}
f(a,b,c,d) & = & a\overline{c}+\overline{a}\,\overline{c}\,\overline{d}+\overline{a}b\overline{c}+bcd+\overline{a}c \\
f(a,b,0,d) & = & a+\overline{a}\,\overline{d}+\overline{a}b \\
f(0,b,0,d) & = & \overline{d}+b \\
f(1,b,0,d) & = & 1 \\
\mathrm{cs}(f(a,b,0,d)) & = & \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1))=a+\overline{d}+b \\
f(a,b,1,d) & = &
\end{array}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c}+\overline{a}\,\overline{c}\,\overline{d}+\overline{a}b\overline{c}+bcd+\overline{a}c \\
f(a,b,0,d) &= a+\overline{a}\,\overline{d}+\overline{a}b \\
f(0,b,0,d) &= \overline{d}+b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd+\overline{a}
\end{aligned}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &=
\end{aligned}$$

Prime Implicants: Example

$$
\begin{aligned}
f(a,b,c,d) &= a\overline{c}+\overline{a}\,\overline{c}\,\overline{d}+\overline{a}b\overline{c}+bcd+\overline{a}c \\
f(a,b,0,d) &= a+\overline{a}\,\overline{d}+\overline{a}b \\
f(0,b,0,d) &= \overline{d}+b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd+\overline{a} \\
f(0,b,1,d) &= 1
\end{aligned}
$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &=
\end{aligned}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &= bd
\end{aligned}$$

Prime Implicants: Example

$$
\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &= bd \\
\mathrm{cs}(f(a,b,1,d)) &=
\end{aligned}
$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &= bd \\
\mathrm{cs}(f(a,b,1,d)) &= \mathrm{abs}((a+1)(\overline{a}+bd)) = \overline{a}+bd
\end{aligned}$$

Prime Implicants: Example

$$
\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &= bd \\
\mathrm{cs}(f(a,b,1,d)) &= \mathrm{abs}((a+1)(\overline{a}+bd)) = \overline{a}+bd \\
\mathrm{cs}(f(a,b,c,d)) &=
\end{aligned}
$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &= bd \\
\mathrm{cs}(f(a,b,1,d)) &= \mathrm{abs}((a+1)(\overline{a}+bd)) = \overline{a}+bd \\
\mathrm{cs}(f(a,b,c,d)) &= \mathrm{abs}((c+a+\overline{d}+b)(\overline{c}+\overline{a}+bd))
\end{aligned}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\mathrm{cs}(f(a,b,0,d)) &= \mathrm{abs}((a+\overline{d}+b)(\overline{a}+1)) = a+\overline{d}+b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &= bd \\
\mathrm{cs}(f(a,b,1,d)) &= \mathrm{abs}((a+1)(\overline{a}+bd)) = \overline{a}+bd \\
\mathrm{cs}(f(a,b,c,d)) &= \mathrm{abs}((c+a+\overline{d}+b)(\overline{c}+\overline{a}+bd)) \\
&= \mathrm{abs}(\overline{a}c + bcd + a\overline{c} + abd + \overline{c}\,\overline{d} + \overline{a}\,\overline{d} + b\overline{c} + \\
&\quad \overline{a}b + bd)
\end{aligned}$$

Prime Implicants: Example

$$\begin{aligned}
f(a,b,c,d) &= a\overline{c} + \overline{a}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c} + bcd + \overline{a}c \\
f(a,b,0,d) &= a + \overline{a}\,\overline{d} + \overline{a}b \\
f(0,b,0,d) &= \overline{d} + b \\
f(1,b,0,d) &= 1 \\
\text{cs}(f(a,b,0,d)) &= \text{abs}((a+\overline{d}+b)(\overline{a}+1)) = a + \overline{d} + b \\
f(a,b,1,d) &= bd + \overline{a} \\
f(0,b,1,d) &= 1 \\
f(1,b,1,d) &= bd \\
\text{cs}(f(a,b,1,d)) &= \text{abs}((a+1)(\overline{a}+bd)) = \overline{a} + bd \\
\text{cs}(f(a,b,c,d)) &= \text{abs}((c+a+\overline{d}+b)(\overline{c}+\overline{a}+bd)) \\
&= \text{abs}(\overline{a}c + bcd + a\overline{c} + abd + \overline{c}\,\overline{d} + \overline{a}\,\overline{d} + b\overline{c} + \\
&\quad \overline{a}b + bd) \\
&= \overline{a}c + a\overline{c} + \overline{c}\,\overline{d} + \overline{a}\,\overline{d} + b\overline{c} + \overline{a}b + bd
\end{aligned}$$

Privileged Cubes

- The transition cubes for each dynamic $1 \longrightarrow 0$ or $0 \longrightarrow 1$ transition are called *priveleged cubes*.
- They cannot be intersected unless the intersecting product also includes its start subcube ($1 \longrightarrow 0$) or end subcube ($0 \longrightarrow 1$).
- If a cover includes a product that intersects a priveleged cube without including its start subcube (or end subcube), then the cover is not hazard-free.

Privileged Cubes: BM Example

		ab			
		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}] \quad 1 \longrightarrow 1$$

$$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd] \quad 0 \longrightarrow 0$$

$$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d] \quad 1 \longrightarrow 0$$

$$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}] \quad 1 \longrightarrow 0$$

Privileged Cubes: BM Example

ab

		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}]$ $1 \longrightarrow 1$ No privileged cubes

$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$ $0 \longrightarrow 0$

$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$ $1 \longrightarrow 0$

$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$ $1 \longrightarrow 0$

Privileged Cubes: BM Example

		ab			
		00	01	11	10
cd	00	1	1	1	1
	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}]$ $1 \longrightarrow 1$ No privileged cubes

$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$ $0 \longrightarrow 0$ No privileged cubes

$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$ $1 \longrightarrow 0$

$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$ $1 \longrightarrow 0$

Privileged Cubes: BM Example

ab

cd	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	1	1	1	0
10	1	1	0	0

$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}]$ $1 \longrightarrow 1$ No privileged cubes

$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$ $0 \longrightarrow 0$ No privileged cubes

$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$ $1 \longrightarrow 0$ $\overline{a}\,\overline{c}$

$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$ $1 \longrightarrow 0$

Privileged Cubes: BM Example

ab

		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}]$ $1 \longrightarrow 1$ No privileged cubes

$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$ $0 \longrightarrow 0$ No privileged cubes

$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$ $1 \longrightarrow 0$ $\overline{a}\,\overline{c}$

$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$ $1 \longrightarrow 0$ c

Privileged Cubes: BM Example

		ab			
		00	01	11	10
	00	1	1	1	1
cd	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

$t_1 = [a\overline{b}\,\overline{c}d, ab\overline{c}\,\overline{d}]$ $1 \longrightarrow 1$ No privileged cubes

$t_2 = [a\overline{b}c\overline{d}, a\overline{b}cd]$ $0 \longrightarrow 0$ No privileged cubes

$t_3 = [\overline{a}b\overline{c}\,\overline{d}, \overline{a}\,\overline{b}\,\overline{c}d]$ $1 \longrightarrow 0$ $\overline{a}\,\overline{c}$

$t_4 = [\overline{a}bcd, a\overline{b}c\overline{d}]$ $1 \longrightarrow 0$ c

priv-set $=$ $\{\overline{a}\,\overline{c}, c\}$

DHF-Prime Implicants

- We may not be able to produce a SOP cover that is free of dynamic hazards using only prime implicants.
- A *dhf-implicant* is an implicant which does not illegally intersect any privileged cube.
- A *dhf-prime implicant* is a dhf-implicant that is contained in no other dhf-implicant.
- A dhf-prime implicant may not be a prime implicant.
- A minimal hazard-free cover includes only dhf-prime implicants.

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	
$\overline{c}\,\overline{d}$	
$b\overline{c}$	
$\overline{a}c$	
$\overline{a}b$	
$\overline{a}\,\overline{d}$	
bd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	
$b\overline{c}$	
$\overline{a}c$	
$\overline{a}b$	
$\overline{a}\,\overline{d}$	
bd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	
$\overline{a}c$	
$\overline{a}b$	
$\overline{a}\,\overline{d}$	
bd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\,\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\,\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}\,c$	
$\overline{a}\,b$	
$\overline{a}\,\overline{d}$	
$b\,d$	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	
$\overline{a}\,\overline{d}$	
bd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	
bd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}\,\overline{c}\,\overline{d}$	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}c\overline{d}$	No, subset of $\overline{c}\,\overline{d}$

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}c\overline{d}$	No, subset of $\overline{c}\,\overline{d}$
abd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}\,\overline{c}\,\overline{d}$	No, subset of $\overline{c}\,\overline{d}$
abd	No, illegally intersects c

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}c\overline{d}$	No, subset of $\overline{c}\,\overline{d}$
abd	No, illegally intersects c
$ab\overline{c}d$	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}\,\overline{c}\,\overline{d}$	No, subset of $\overline{c}\,\overline{d}$
abd	No, illegally intersects c
$ab\overline{c}d$	No, subset of $a\overline{c}$

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}\,\overline{c}\,\overline{d}$	No, subset of $\overline{c}\,\overline{d}$
abd	No, illegally intersects c
$ab\overline{c}d$	No, subset of $a\overline{c}$
bcd	

DHF-Prime Implicants: Example

Privileged Cube	Start Subcube
$\overline{a}\,\overline{c}$	$\overline{a}b\overline{c}\,\overline{d}$
c	$\overline{a}bcd$

Primes	DHF-Prime
$a\overline{c}$	Yes
$\overline{c}\,\overline{d}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$b\overline{c}$	Yes, legally intersects $\overline{a}\,\overline{c}$
$\overline{a}c$	Yes, legally intersects c
$\overline{a}b$	Yes, legally intersects both
$\overline{a}\,\overline{d}$	No, illegally intersects c
bd	No, illegally intersects $\overline{a}\,\overline{c}$
$\overline{a}\,\overline{c}\,\overline{d}$	No, subset of $\overline{c}\,\overline{d}$
abd	No, illegally intersects c
$ab\overline{c}d$	No, subset of $a\overline{c}$
bcd	Yes

Setting up the Covering Problem

$$
\begin{array}{lcccccc}
 & \overline{a}c & a\overline{c} & \overline{c}\overline{d} & b\overline{c} & \overline{a}b & bcd \\
a\overline{c} & - & 1 & - & - & - & - \\
\overline{a}\overline{c}\overline{d} & - & - & 1 & - & - & - \\
\overline{a}b\overline{c} & - & - & - & 1 & 1 & - \\
bcd & - & - & - & - & - & 1 \\
\overline{a}c & 1 & - & - & - & - & -
\end{array}
$$

Setting up the Covering Problem

$$\begin{array}{ccccccc}
 & \overline{a}c & a\overline{c} & \overline{c}\,\overline{d} & b\overline{c} & \overline{a}b & bcd \\
a\overline{c} & - & 1 & - & - & - & - \\
\overline{a}\,\overline{c}\,\overline{d} & - & - & 1 & - & - & - \\
\overline{a}b\overline{c} & - & - & - & 1 & 1 & - \\
bcd & - & - & - & - & - & 1 \\
\overline{a}c & 1 & - & - & - & - & -
\end{array}$$

$$f = \overline{a}c + a\overline{c} + \overline{c}\,\overline{d} + b\overline{c} + bcd$$

Generalized Transition Cube

- *Generalized transition cube* allows start and end points to be cubes rather than simply minterms.
- In the *generalized transition cube* $[c_1, c_2]$, the cube c_1 is called the *start cube* and c_2 is called the *end cube*.
- The *open generalized transition cube*, $[c_1, c_2)$, is all minterms in $[c_1, c_2]$ excluding those in c_2 (i.e., $[c_1, c_2) = [c_1, c_2] - c_2$).

Extended Burst-Mode Transitions

- In XBM machine, some signals are rising, some are falling, and others are levels which can change nonmonotonically.
- Rising and falling signals change monotonically.
- Level signals must hold the same value in c_1 and c_2, where the value is either a constant (0 or 1) or a don't care ($-$).
- Level signals may change nonmonotonically.
- Transitions are restricted such that each function may change value only after the completion of an input burst.
- $[c_1, c_2]$ for a function f is an *extended burst-mode transition* if for every minterm $m_i \in [c_1, c_2)$, $f(m_i) = f(c_1)$ and for every minterm $m_i \in c_2$, $f(m_i) = f(c_2)$.
- If a function has only extended burst-mode transitions, then it is function hazard-free.

Extended Burst-Mode Transitions: Example

		xl			
		00	01	11	10
yz	00	1	1	1	1
	01	1	1	1	1
	11	1	1	0	0
	10	1	1	0	0

$$[\overline{x}-\overline{y}-, x-y-]$$

Extended Burst-Mode Transitions: Example

		xl			
		00	01	11	10
yz	00	1	1	1	1
	01	1	1	1	1
	11	1	1	0	0
	10	1	1	0	0

$$[\overline{x}-\overline{y}-, x-y-]$$

Extended burst-mode transition

Extended Burst-Mode Transitions: Example

		xl			
		00	01	11	10
	00	1	1	1	1
yz	01	1	1	1	1
	11	1	1	0	0
	10	1	1	0	0

$$[\overline{x} - \overline{y} -, x - y\, z]$$

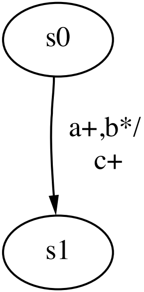
Extended Burst-Mode Transitions: Example

		xl			
		00	01	11	10
yz	00	1	1	1	1
	01	1	1	1	1
	11	1	1	0	0
	10	1	1	0	0

$$[\overline{x} - \overline{y} -, x - y\, z]$$

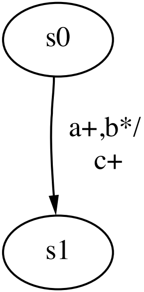
Not an extended burst-mode transition

Extended Burst-Mode to Flow Table



	ab 00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s1,1	01
s1			s1,1	s1,1	10

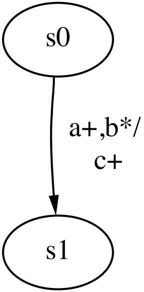
Extended Burst-Mode to Flow Table



	ab				
	00	01	11	10	xy
s0	s0,0	s0,0	s1,1	s1,1	01
s1			s1,1	s1,1	10

$$[\overline{a} - \overline{x}y, a - \overline{x}y]$$

Extended Burst-Mode to Flow Table

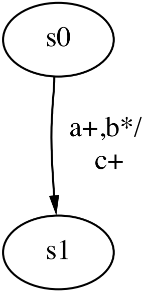


	ab 00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s1,1	01
s1			s1,1	s1,1	10

$$[\overline{a} - \overline{x}y, a - \overline{x}y]$$

Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Extended Burst-Mode to Flow Table



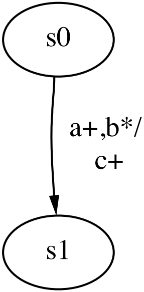
	ab				
	00	01	11	10	xy
s0	s0,0	s0,0	s1,1	s1,1	01
s1			s1,1	s1,1	10

$$[\overline{a} - \overline{x}y, a - \overline{x}y]$$

Dynamic 0 → 1 transition for output c and next-state variable X

Dynamic 1 → 0 transition for next-state variable Y

Extended Burst-Mode to Flow Table



	ab				
	00	01	11	10	xy
s0	s0,0	s0,0	s1,1	s1,1	01
s1			s1,1	s1,1	10

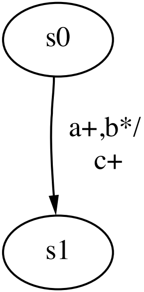
$$[\overline{a} - \overline{x}y, a - \overline{x}y]$$

Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Dynamic 1 → 0 transition for next-state variable *Y*

$$[a - \overline{x}y, a - x\overline{y}]$$

Extended Burst-Mode to Flow Table



	ab				
	00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s1,1	01
s1			s1,1	s1,1	10

$$[\overline{a}-\overline{x}y, a-\overline{x}y]$$

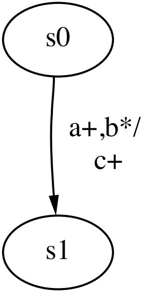
Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Dynamic 1 → 0 transition for next-state variable *Y*

$$[a-\overline{x}y, a-x\overline{y}]$$

Static 1 → 1 transition for output *c* and next-state variable *X*

Extended Burst-Mode to Flow Table



	ab				
	00	01	11	10	*xy*
s0	s0,0	s0,0	s1,1	s1,1	01
s1			s1,1	s1,1	10

$$[\overline{a} - \overline{x}y, a - \overline{x}y]$$

Dynamic 0 → 1 transition for output *c* and next-state variable *X*

Dynamic 1 → 0 transition for next-state variable *Y*

$$[a - \overline{x}y, a - x\overline{y}]$$

Static 1 → 1 transition for output *c* and next-state variable *X*

Static 0 → 0 transition for next-state variable *Y*

Start and End Subcubes

- *Start subcube*, c_1', is maximal subcube of c_1 where signals having directed don't-care transitions are set to initial value.
- *End subcube*, c_2', is maximal subcube of c_2 where signals having directed don't-care transitions are set to final value.

Start and End Subcube: Example

xl

		00	01	11	10
	00	1	1	1	1
yz	01	1	1	1	1
	11	1	1	0	0
	10	1	1	0	0

$$[\overline{x} - \overline{y} -, x - y -]$$

Assume that *z* is a rising directed don't care transition.

Start and End Subcube: Example

		xl			
		00	01	11	10
yz	00	1	1	1	1
	01	1	1	1	1
	11	1	1	0	0
	10	1	1	0	0

$$[\overline{x} - \overline{y} -, x - y -]$$

Assume that z is a rising directed don't care transition.

$\overline{x} - \overline{y}\,\overline{z}$ is the start subcube.

Start and End Subcube: Example

		xl			
		00	01	11	10
	00	1	1	1	1
yz	01	1	1	1	1
	11	1	1	0	0
	10	1	1	0	0

$$[\overline{x} - \overline{y} -, x - y -]$$

Assume that z is a rising directed don't care transition.

$\overline{x} - \overline{y}\,\overline{z}$ is the start subcube.

$x - yz$ is the end subcube.

Hazard Issues

- Considering $[c'_1, c'_2]$, hazard considerations are same.
- If a static 1 → 1 transition the entire transition cube must be included in some product term in the cover.
- If a dynamic 1 → 0 transition, any product that intersects this transition cube must contain the start subcube, c'_1.
- Must also consider dynamic 0 → 1 transitions.
- Any product that intersects transition cube for a dynamic 0 → 1 transition must contain the end subcube, c'_2.

XBM Hazard Issues: Example

		xl			
		00	01	11	10
	00	0	0	0	0
yz	01	0	0	1	1
	11	0	0	1	0
	10	0	0	0	0

$$[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$$

XBM Hazard Issues: Example

		xl			
		00	01	11	10
yz	00	0	0	0	0
	01	0	0	1	1
	11	0	0	1	0
	10	0	0	0	0

$$[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$$

Dynamic 0 → 1 transition

XBM Hazard Issues: Example

		xl			
		00	01	11	10
	00	0	0	0	0
yz	01	0	0	1	1
	11	0	0	1	0
	10	0	0	0	0

$$[\overline{x}\ -\ \overline{y}\,\overline{z}, x\ -\ \overline{y}\,z]$$

Dynamic 0 → 1 transition

$$[\overline{x}\,l\,y\,\overline{z}, x\,l\,y\,z]$$

XBM Hazard Issues: Example

		xl			
		00	01	11	10
yz	00	0	0	0	0
	01	0	0	1	1
	11	0	0	1	0
	10	0	0	0	0

$$[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$$

Dynamic 0 → 1 transition

$$[\overline{x}\,l\,y\,\overline{z}, x\,l\,y\,z]$$

Dynamic 0 → 1 transition

XBM Hazard Issues: Example

xl

		00	01	11	10
	00	0	0	0	0
yz	01	0	0	1	1
	11	0	0	1	0
	10	0	0	0	0

$$[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$$

Dynamic 0 → 1 transition

$$[\overline{x}\,l\,y\,\overline{z}, x\,l\,y\,z]$$

Dynamic 0 → 1 transition

$$f = x\overline{y}z + xlz$$

XBM Hazard Issues: Example

		xl 00	01	11	10
	00	0	0	0	0
yz	01	0	0	1	1
	11	0	0	1	0
	10	0	0	0	0

$$[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$$

Dynamic 0 → 1 transition

$$[\overline{x}\,l\,y\,\overline{z}, x\,l\,y\,z]$$

Dynamic 0 → 1 transition

$$f = x\overline{y}z + xlz$$

xlz illegally intersects $[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$

XBM Hazard Issues: Example

		xl			
		00	01	11	10
yz	00	0	0	0	0
	01	0	0	1	1
	11	0	0	1	0
	10	0	0	0	0

$$[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$$

Dynamic 0 → 1 transition

$$[\overline{x}\,l\,y\,\overline{z}, x\,l\,y\,z]$$

Dynamic 0 → 1 transition

$$f = x\overline{y}z + xlz$$

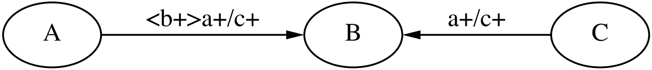
xlz illegally intersects $[\overline{x} - \overline{y}\,\overline{z}, x - \overline{y}\,z]$

Must reduce to: $f = x\overline{y}z + xlyz$

DHF-Compatibles

- Two states s_1 and s_2 are *dhf-compatible* when they are compatible and for each output z and transition $[c_1, c_2]$ of s_1 and for each transition $[c_3, c_4]$ of s_2:
  1. If z has a $1 \rightarrow 0$ transition in $[c_1, c_2]$ and a $1 \rightarrow 1$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$ or $c_1' \in [c_3, c_4]$.
  2. If z has a $1 \rightarrow 0$ transition in $[c_1, c_2]$ and a $1 \rightarrow 0$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$, $c_1 = c_3$, $[c_1, c_2] \subseteq [c_3, c_4]$, or $[c_3, c_4] \subseteq [c_1, c_2]$.
  3. If z has a $0 \rightarrow 1$ transition in $[c_1, c_2]$ and a $1 \rightarrow 1$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$ or $c_2' \in [c_3, c_4]$.
  4. If z has a $0 \rightarrow 1$ transition in $[c_1, c_2]$ and a $0 \rightarrow 1$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$, $c_2 = c_4$, $[c_1, c_2] \subseteq [c_3, c_4]$, or $[c_3, c_4] \subseteq [c_1, c_2]$.

State Minimization: Extended Burst-Mode



Inputs a b

State		00	01	11	10
	A	A,0	A,0	B,1	D,0
	B	–	–	B,1	B,1
	C	C,0	C,0	B,1	B,1

Inputs a b

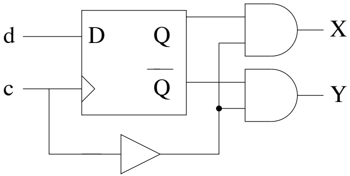
State		00	01	11	10
	A	A,0	A,0	B,1	D,0
	BC	B,0	B,0	B,1	B,1

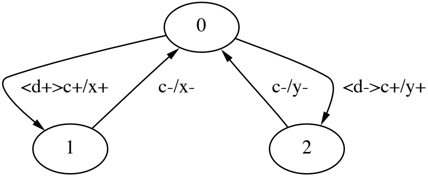
For input 11, static 1 → 1 transition when transition from A to B.
In state BC, dynamic 0 → 1 transition.

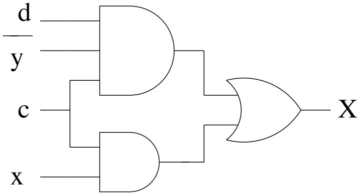
Further Restrictions

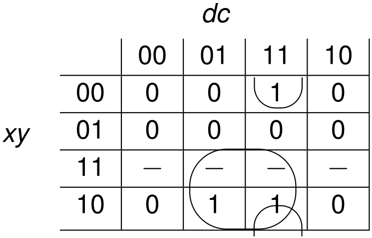
- s_1 and s_2 must also satisfy the following further restriction for each s_3, which can transition to s_1 in $[c_3, c_4]$ and another transition $[c_1, c_2]$ of s_2:
  1. If z has a $1 \rightarrow 0$ transition in $[c_1, c_2]$ and a $1 \rightarrow 1$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$ or $c_1' \in [c_3, c_4]$.
  2. If z has a $0 \rightarrow 1$ transition in $[c_1, c_2]$ and a $1 \rightarrow 1$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$ or $c_2' \in [c_3, c_4]$.
- For each s_3 which can transition to s_2 in $[c_3, c_4]$ and another transition $[c_1, c_2]$ of s_1:
  1. If z has a $1 \rightarrow 0$ transition in $[c_1, c_2]$ and a $1 \rightarrow 1$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$ or $c_1' \in [c_3, c_4]$.
  2. If z has a $0 \rightarrow 1$ transition in $[c_1, c_2]$ and a $1 \rightarrow 1$ transition in $[c_3, c_4]$, then $[c_1, c_2] \cap [c_3, c_4] = \emptyset$ or $c_2' \in [c_3, c_4]$.

Extended Burst-Mode Dynamic Hazard Problem









dc

xy

	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	—	—	—	—
10	0	1	1	0

Static 1 → 1 transition $[dc\overline{x}\,\overline{y}, dcx\overline{y}]$

Dynamic 1 → 0 transition $[-cx\overline{y}, -\overline{c}x\overline{y}]$

State Assignment

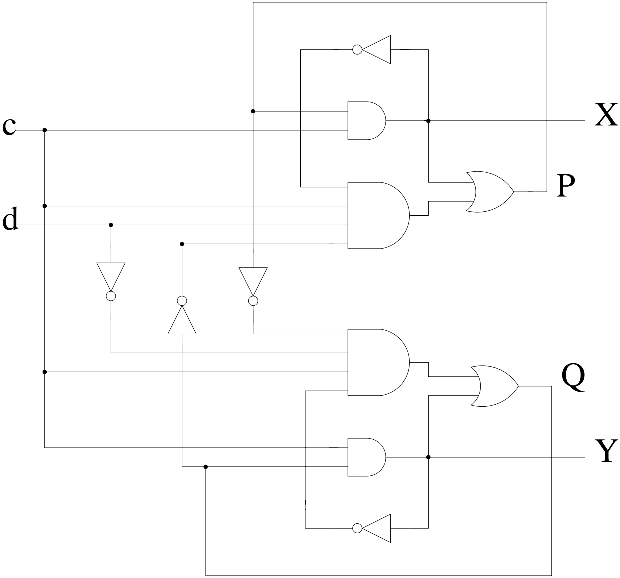
		dc			
		00	01	11	10
	000	0	0	0	0
	001	—	—	—	—
	011	—	—	—	—
pxy	010	—	—	—	—
	110	0	1	1	0
	111	—	—	—	—
	101	—	—	—	—
	100	0	1	1	0

X

		dc			
		00	01	11	10
	000	0	0	1	0
	001	—	—	—	—
	011	—	—	—	—
pxy	010	—	—	—	—
	110	0	1	1	0
	111	—	—	—	—
	101	—	—	—	—
	100	0	1	1	0

P

Hazard-free dff Circuit



XBM Example

		abc							
		000	001	011	010	110	111	101	100
	00	0	0	0	0	0	0	0	0
de	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$				
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$				
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \longrightarrow 1$			
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$				
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

abc

		000	001	011	010	110	111	101	100
	00	0	0	0	0	0	0	0	0
de	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a} b d]$	$0 \rightarrow 1$	$\overline{a} b d$		
$[a b \overline{c} d \overline{e}, a b c d \overline{e}]$				
$[a \overline{b} d, a \overline{b}\,\overline{d}]$				

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	0 → 1	$\overline{a}\,b\,d$	$\overline{a}$	
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$				
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \longrightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$				
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

		abc							
		000	001	011	010	110	111	101	100
	00	0	0	0	0	0	0	0	0
de	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$			
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$		
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	0 → 1	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	1 → 1	$a\,b\,d\,\overline{e}$	none	
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	0 → 1	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	1 → 1	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$				

XBM Example

abc

		000	001	011	010	110	111	101	100
	00	0	0	0	0	0	0	0	0
de	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$	$1 \rightarrow 0$			

XBM Example

abc

		000	001	011	010	110	111	101	100
	00	0	0	0	0	0	0	0	0
de	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}bd]$	$0 \longrightarrow 1$	$\overline{a}bd$	$\overline{a}$	$\overline{a}b\overline{c}d$ (end)
$[ab\overline{c}d\overline{e}, abcd\overline{e}]$	$1 \longrightarrow 1$	$abd\overline{e}$	none	none
$[a\overline{b}d, a\overline{b}\,\overline{d}]$	$1 \longrightarrow 0$	$a\overline{b}d$		

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$	$1 \rightarrow 0$	$a\,\overline{b}\,d$	$a\,\overline{b}$	

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$	$1 \rightarrow 0$	$a\,\overline{b}\,d$	$a\,\overline{b}$	$a\,\overline{b}\,\overline{c}\,d$ (start)

XBM Example

abc

		000	001	011	010	110	111	101	100
	00	0	0	0	0	0	0	0	0
de	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$	$1 \rightarrow 0$	$a\,\overline{b}\,d$	$a\,\overline{b}$	$a\,\overline{b}\,\overline{c}\,d$ (start)

$$\text{primes} = \{\overline{a}\,b\,d, a\,\overline{b}\,d, a\,d\,\overline{e}, b\,d\,\overline{e}\}$$

XBM Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$	$1 \rightarrow 0$	$a\,\overline{b}\,d$	$a\,\overline{b}$	$a\,\overline{b}\,\overline{c}\,d$ (start)

$$\text{primes} = \{\overline{a}\,b\,d, a\,\overline{b}\,d, a\,d\,\overline{e}, b\,d\,\overline{e}\}$$
$$\text{DHF-primes} = \{\overline{a}\,b\,d, a\,\overline{b}\,d, a\,b\,d\,\overline{e}\}$$

XBM Example

abc

		000	001	011	010	110	111	101	100
	00	0	0	0	0	0	0	0	0
de	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a} b d]$	$0 \rightarrow 1$	$\overline{a} b d$	$\overline{a}$	$\overline{a} b \overline{c} d$ (end)
$[a b \overline{c} d \overline{e}, a b c d \overline{e}]$	$1 \rightarrow 1$	$a b d \overline{e}$	none	none
$[a \overline{b} d, a \overline{b}\,\overline{d}]$	$1 \rightarrow 0$	$a \overline{b} d$	$a \overline{b}$	$a \overline{b}\,\overline{c} d$ (start)

$$\text{primes} = \{\overline{a} b d, a \overline{b} d, a d \overline{e}, b d \overline{e}\}$$

$$\text{DHF-primes} = \{\overline{a} b d, a \overline{b} d, a b d \overline{e}\}$$

$$f = \overline{a} b d + a \overline{b} d + a b d \overline{e}$$

Multi-Level Logic Synthesis

- Two-level SOP implementations cannot be realized directly for most technologies.
- AND or OR stages of arbitrarily large fan-in not practical.
- In CMOS, gates with more than 3 or 4 inputs are too slow.
- Two-level SOP implementations must be decomposed using Boolean algebra laws into multi-level implementations.
- Care must be taken not to introduce hazards.
- We present a number of *hazard-preserving transformations*.
- If we begin with a hazard-free SOP implementation and only apply hazard-preserving transformations than the resulting multi-level implementation is also hazard-free.

Hazard-Preserving Transformations

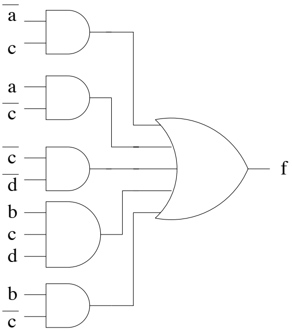
- **Theorem 5.19** (Unger, 1969) Given any expression f_1, if we transform it into another expression, f_2, using the following laws:
  - $A + (B + C) \Leftrightarrow A + B + C$ (associative law)
  - $A(BC) \Leftrightarrow ABC$ (associative law)
  - $\overline{(A + B)} \Leftrightarrow \overline{A}\,\overline{B}$ (DeMorgan's theorem)
  - $\overline{(AB)} \Leftrightarrow \overline{A} + \overline{B}$ (DeMorgan's theorem)
  - $AB + AC \Rightarrow A(B + C)$ (distributive law)
  - $A + AB \Rightarrow A$ (absorptive law)
  - $A + \overline{A}B \Rightarrow A + B$

  then a circuit corresponding to f_2 will have no combinational hazards not present in circuits corresponding to f_1.

More Hazard-Preserving Transformations

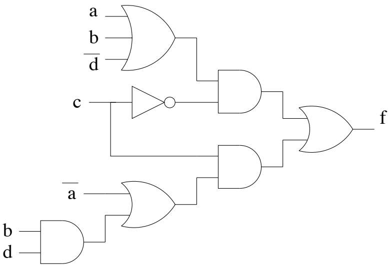
- Hazard exchanges:
  - Insertion or deletion of inverters at the output of a circuit only interchanges 0 and 1-hazards.
  - Insertion or deletion of inverters at the inputs only relocates hazards to duals of original transition.
  - The dual of a circuit (exchange AND and OR gates) produces dual function with dual hazards.

Multilevel Logic Synthesis: Example



$$f = \overline{a}c + a\overline{c} + \overline{c}\overline{d} + bcd + b\overline{c}$$

Multilevel Logic Synthesis: Example



$$f = \overline{c}(a+b+\overline{d})+c(\overline{a}+bd)$$

Technology Mapping

- *Technology mapping* step takes as input a set of technology-independent logic equations and a library of cells, and it produces a netlist of cells.
- Broken up into three major steps:
  - *Decomposition*,
  - *Partitioning*, and
  - *Matching/covering*.

Decomposition

- Decomposition transforms logic equations into equivalent network using only two-input/one-output *base functions*.
- A typical choice of base function is two-input NAND gates.
- Decomposition performed using recursive applications of DeMorgan's theorem and the associative law.
- These operations are hazard-preserving.
- Simplification during this step may remove redundant logic added to eliminate hazards, so must be avoided.

Decomposition Example

$$f = \overline{c}(a+b+\overline{d})+c(\overline{a}+bd)$$

Decomposition Example

$$
\begin{aligned}
f &= \overline{c}(a+b+\overline{d})+c(\overline{a}+bd) \\
f &= \overline{c}((a+b)+\overline{d})+c(\overline{a}+bd) \text{ (associative law)}
\end{aligned}
$$

Decomposition Example

$$\begin{aligned} f &= \bar{c}(a+b+\bar{d}) + c(\bar{a}+bd) \\ f &= \bar{c}((a+b)+\bar{d}) + c(\bar{a}+bd) \text{ (associative law)} \\ f &= \bar{c}(\overline{(\bar{a}\bar{b})}+\bar{d}) + c(\bar{a}+bd) \text{ (DeMorgan's theorem)} \end{aligned}$$

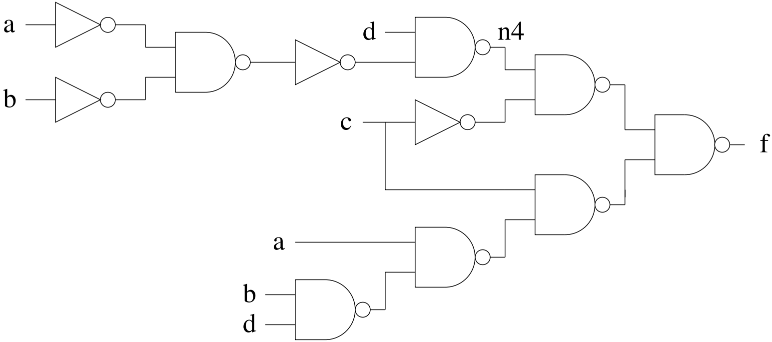
Decomposition Example

$$
\begin{aligned}
f &= \overline{c}\,(a+b+\overline{d}) + c\,(\overline{a}+b\,d)\\
f &= \overline{c}\,((a+b)+\overline{d}) + c\,(\overline{a}+b\,d) \text{ (associative law)}\\
f &= \overline{c}\,(\overline{(\overline{a}\,\overline{b})}+\overline{d}) + c\,(\overline{a}+b\,d) \text{ (DeMorgan's theorem)}\\
f &= \overline{c}\,\overline{((\overline{a}\,\overline{b})\,d)} + c\,\overline{(a\,\overline{(b\,d)})} \text{ (DeMorgan's theorem)}
\end{aligned}
$$

Decomposition Example

$$
\begin{aligned}
f &= \overline{c}\,(a+b+\overline{d})+c\,(\overline{a}+b\,d) \\
f &= \overline{c}\,((a+b)+\overline{d})+c\,(\overline{a}+b\,d) \text{ (associative law)} \\
f &= \overline{c}\,(\overline{(\overline{a}\,\overline{b})}+\overline{d})+c\,(\overline{a}+b\,d) \text{ (DeMorgan's theorem)} \\
f &= \overline{c}\,\overline{((\overline{a}\,\overline{b})\,d)}+c\,\overline{(a\,\overline{(b\,d)})} \text{ (DeMorgan's theorem)} \\
f &= \overline{\overline{(\overline{c}\,\overline{((\overline{a}\,\overline{b})\,d)})}\,\overline{(c\,\overline{(a\,\overline{(b\,d)})})}} \text{ (DeMorgan's theorem)}
\end{aligned}
$$

Decomposition Example



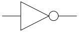
Partitioning

- Partitioning breaks up decomposed network at points of multiple fanout into single output cones of logic.
- Since partitioning step does not change the topology of the network, it does not affect the hazard behavior of the network.

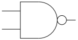
Matching and Covering

- Matching and covering examines each cone of logic and finds cells in the library to implement subnetworks within the cone.
- Can be implemented either using *structural pattern-matching* or *Boolean matching* techniques.
- In the structural techniques, each library element is also decomposed into base functions.
- Library elements are then compared against portions of the network to be mapped using pattern matching.
- Assuming that the decomposed logic and library gates are hazard-free, the resulting mapped logic is also hazard-free.

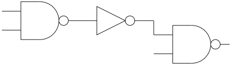
Gate Library



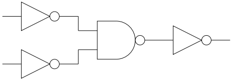
Inv(Cost = 1)



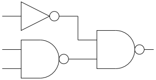
2NAND(Cost = 3)



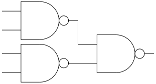
3NAND(Cost = 5)



2NOR(Cost = 2)

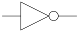


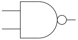
AOI1(Cost = 3)

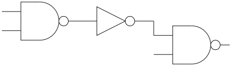


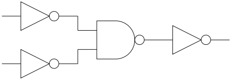
AOI2(Cost = 4)

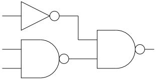
Matching and Covering Example

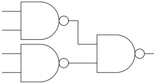

Inv(Cost = 1)

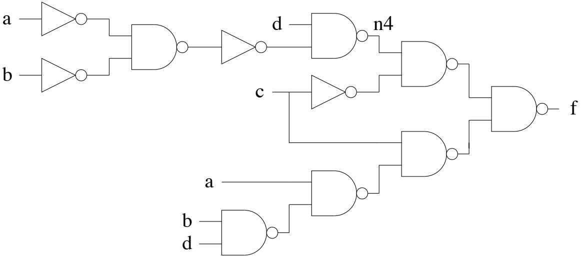

2NAND(Cost = 3)


3NAND(Cost = 5)

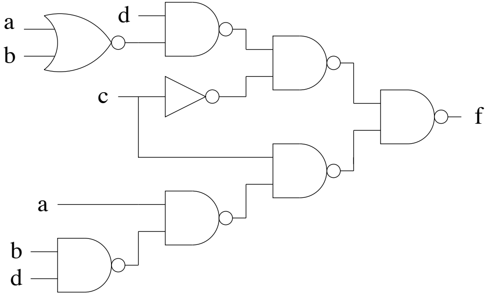

2NOR(Cost = 2)


AOI1(Cost = 3)

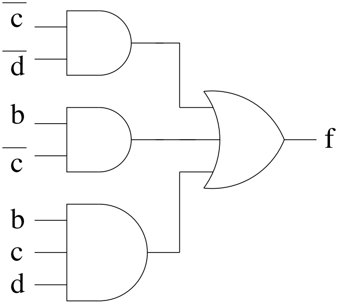

AOI2(Cost = 4)

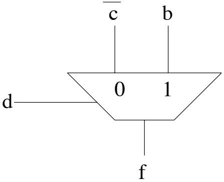


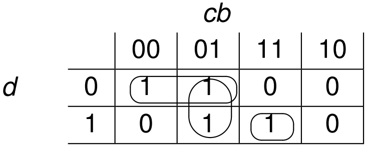
Final Mapped Circuit



Boolean Matching

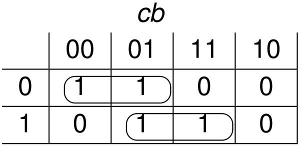






cb

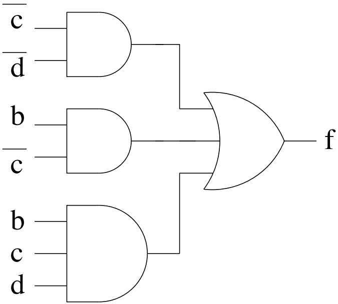
		00	01	11	10
d	0	1	1	0	0
	1	0	1	1	0

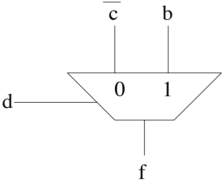


cb

	00	01	11	10
0	1	1	0	0
1	0	1	1	0

Boolean Matching





cb

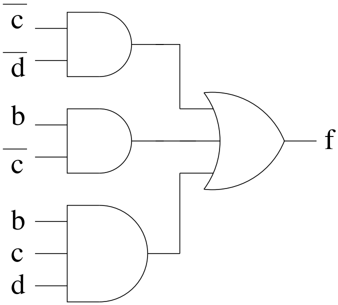
		00	01	11	10
d	0	1	1	0	0
	1	0	1	1	0

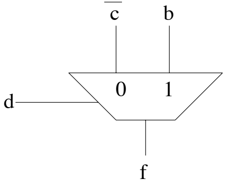
cb

	00	01	11	10
0	1	1	0	0
1	0	1	1	0

Dynamic 1 → 0 transition $[\bar{a}b\bar{c}\bar{d}, \bar{a}\bar{b}\bar{c}d]$

Boolean Matching





cb

d	00	01	11	10
0	1	1	0	0
1	0	1	1	0

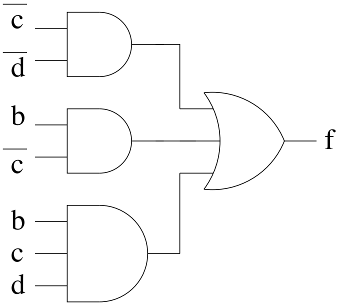
cb

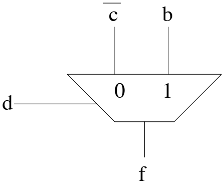
d	00	01	11	10
0	1	1	0	0
1	0	1	1	0

Dynamic 1 → 0 transition $[\overline{a}b\overline{c}\overline{d}, \overline{a}\overline{b}\overline{c}d]$

Multiplexor has a dynamic 1 → 0 hazard.

Boolean Matching





		cb			
		00	01	11	10
d	0	1	1	0	0
	1	0	1	1	0

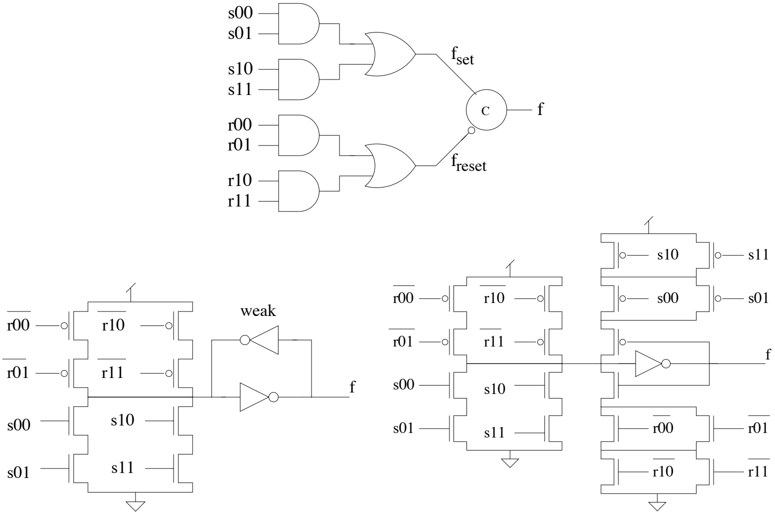
	cb			
	00	01	11	10
0	1	1	0	0
1	0	1	1	0

Dynamic 1 → 0 transition $[\overline{a}\, b\, \overline{c}\, \overline{d}, \overline{a}\, \overline{b}\, \overline{c}\, d]$

Multiplexor has a dynamic 1 → 0 hazard.

If original implementation was $f = \overline{c}\,\overline{d} + bd$ then multiplexor would be okay.

Generalized C-Elements



Generalized C-Element Hazard Issues

- Static hazards cannot manifest on the output of a gC gate.
- Prolonged short-circuit current should be avoided.
- Decomposition of trigger signals which during a transition both enable and disable a P and N stack is not allowed.
- By avoiding short circuits, product terms intersecting a dynamic transition no longer must include the start subcube.
- The problems with conditionals and dynamic hazards are also not present in gC implementations.

Hazard Requirements

- The hazard-free cover requirements for the set function, f_{set}, in an extended burst-mode gC become:

  1. Each set cube of f_{set} must not include OFF-set minterms.
  2. For every dynamic $0 \longrightarrow 1$ transition $[c_1, c_2]$ in f_{set}, the end cube, c_2, must be covered by some product term.
  3. Any product of f_{set} intersecting c_2 of a dynamic $0 \longrightarrow 1$ transition $[c_1, c_2]$ must also contain the end subcube c_2'.

- Hazard-freedom requirements for f_{reset} are analogous to f_{set}.

Generalized C-Element Example

		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$	$1 \rightarrow 0$	$a\,\overline{b}\,d$	$a\,\overline{b}$	$a\,\overline{b}\,\overline{c}\,d$ (start)

Generalized C-Element Example

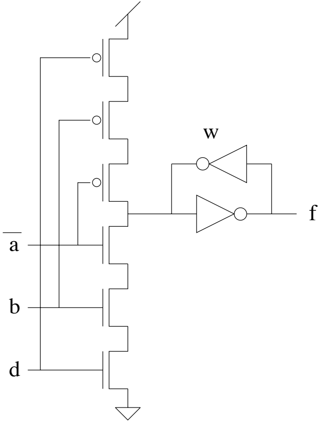
		abc							
		000	001	011	010	110	111	101	100
de	00	0	0	0	0	0	0	0	0
	01	0	0	0	0	0	0	0	0
	10	0	0	1	1	1	1	1	1
	11	0	0	1	1	0	0	1	1

Transition Cube	Type	Required Cube	Privileged Cube	Subcube
$[\overline{a}\,\overline{b}\,\overline{d}, \overline{a}\,b\,d]$	$0 \rightarrow 1$	$\overline{a}\,b\,d$	$\overline{a}$	$\overline{a}\,b\,\overline{c}\,d$ (end)
$[a\,b\,\overline{c}\,d\,\overline{e}, a\,b\,c\,d\,\overline{e}]$	$1 \rightarrow 1$	$a\,b\,d\,\overline{e}$	none	none
$[a\,\overline{b}\,d, a\,\overline{b}\,\overline{d}]$	$1 \rightarrow 0$	$a\,\overline{b}\,d$	$a\,\overline{b}$	$a\,\overline{b}\,\overline{c}\,d$ (start)

Only need to consider the two dynamic transitions.

Generalized C-Element Example

$$f_{\text{set}} = \overline{a} b d$$
$$f_{\text{reset}} = a \overline{b}\, \overline{d}$$



Sequential Hazards

- Huffman circuits require that outputs and state variables stabilize before either new inputs or fed-back state variables arrive.
- A violation of this assumption can result in a *sequential hazard*.
- Presence of a sequential hazard is dependent on timing of the environment, circuit, and feedback delays.

Essential Hazard

	x	
	0	1
1	①,0	2,0
2	3,1	②,0
3	③,1	k,1

Feedback Delay Requirement

- To eliminate essential hazards, there is a *feedback delay requirement*:

  $$D_f \quad \geq \quad d_{\text{max}} - d_{\text{min}}$$

  where D_f is the feedback delay, d_{max} is the maximum delay in the combinational logic, and d_{min} is the minimum delay through the combinational logic.

Fundamental-Mode Constraint

- Sequential hazards can also result if the environment reacts too quickly.
- Fundamental-mode environmental constraint says inputs are not allowed to change until the circuit stabilizes.
- To satisfy this constraint, a conservative separation time needed between inputs can be expressed as follows:

  $$d_i \geq 2d_{\mathrm{max}} + D_f$$

  where d_i is the separation time needed between input bursts.
- Separation needs a $2d_{\mathrm{max}}$ term since the circuit must respond to the input change followed by the subsequent state change.

Setup and Hold Time Constraint

- XBM machines require a *setup time* and *hold time* for conditional signals.
- Conditional signals must stabilize a setup time before the compulsory signal transition which samples them.
- It must remain stable a hold time after the output and state changes complete.
- Outside this window of time, the conditional signals are free to change arbitrarily.

Summary

- Binate covering problems
- State minimization
- State assignment
- Hazard-free logic synthesis
- Extensions for MIC operation
- Multilevel logic synthesis
- Technology mapping
- Generalized C-element implementation
- Sequential hazards