Combinational Problems: Unate Covering, Binate Covering, Graph Coloring and Maximum Cliques

Example of application: Decomposition

What will be discussed

- In many logic synthesis, formal verification or testing problems we can reduce the problem to some well-known and researched problem
- These problems are called combinational problems or discrete optimization problems
- In our area of interest the most often used are the following:
  - Set Covering (Unate Covering)
  - Covering/Closure (Binate Covering)
  - Graph Coloring
  - Maximum Clique

Basic Combinational Problems

- There are many problems that can be reduced to Graph Coloring.
- They include
  - Two-Level (SOP) minimization
  - Three-Level minimization
  - Minimum test set
  - Boolean and Multiple-valued Decomposition

Basic Combinational Problems

- There are many problems that can be reduced to ***Unate Covering***
- They include
  - Two-Level (SOP) minimization
  - Three-Level CDEC minimization (Conditional DECoder)
  - Minimum test set generation
  - Boolean and Multiple-valued Decomposition
  - Physical and technology mapping

Basic Combinational Problems

- There are many problems that can be reduced to ***Binate Covering***
- They include
  - Three-Level TANT minimization (Three Level AND NOT Networks with True Inputs)
  - FSM state minimization (Finite State Machine)
  - Technology mapping
- Binate Covering is basically the same as Satisfiability that has hundreds of applications

Ashenhurst/Curtis Decomposition

- Ashenhurst created a method to decompose a single-output Boolean function to sub-functions
- Curtis generalized this method to decompositions with more than one wire between the subfunctions.
- Miller and Muzio generalized to Multi-Valued Logic functions
- Perkowski et al generalized to Relations

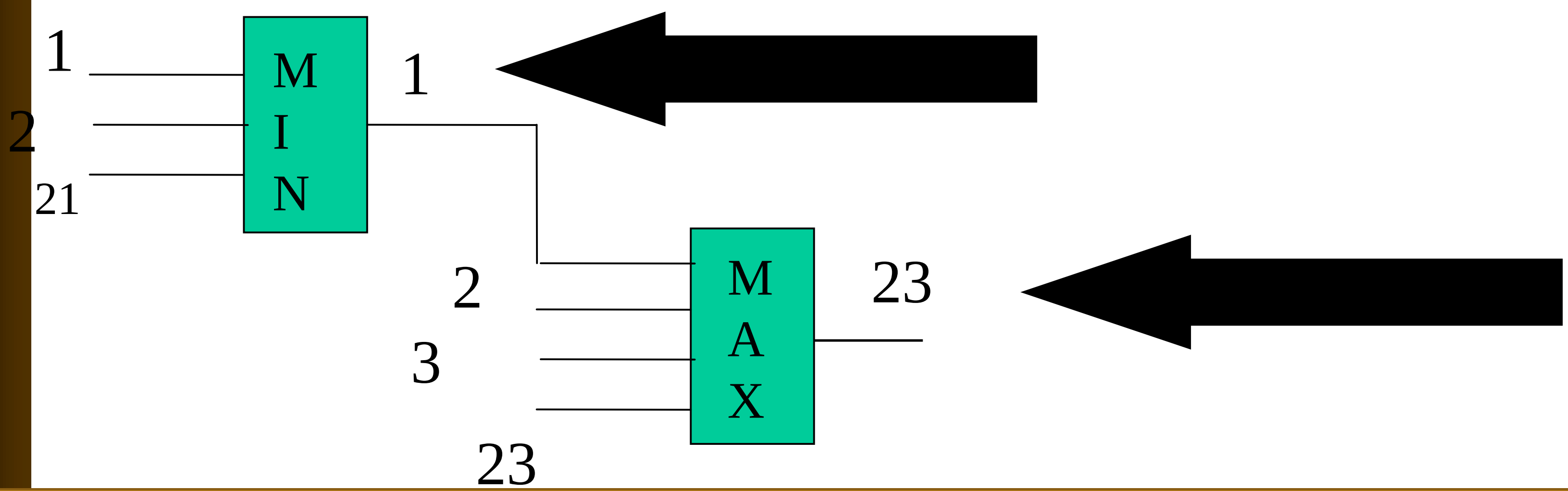
Short Introduction: multiple-valued logic

Signals can have values from some set, for instance {0,1,2}, or {0,1,2,3}

{0,1} - binary logic (a special case)

{0,1,2} - ***a ternary logic***

{0,1,2,3} ***- a quaternary logic,*** **etc**



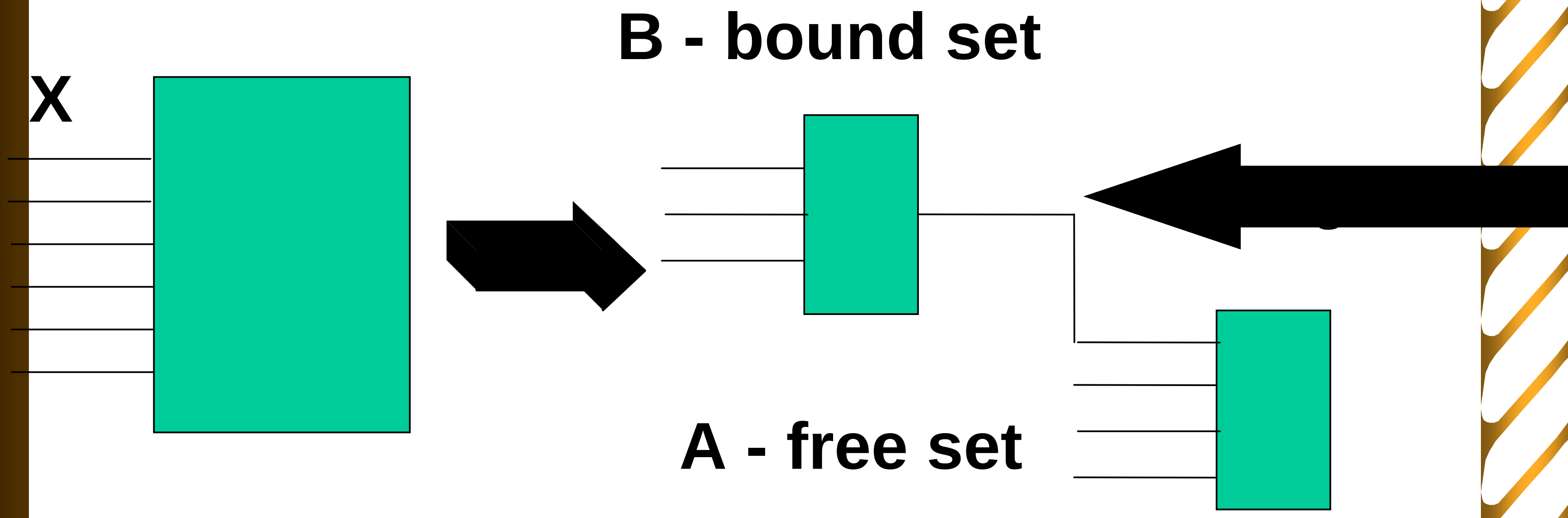
Multiple-valued logic relations

AB \ CD	00	01	11	10
00	0 or 1	1 or 2	0,1,or2	0
01	0	2 or 3	0,1,or2	1
11	1	1 or 2	0,1,or2	3
10	2	0,1,or2	0	3

Example of a relation with 4 binary inputs and one 4-valued (quaternary) output variable

Two-Level Ashenhurst Decomposition

$$F(X) = H(\,G(B),\, A\,),\quad X = A \cup B$$

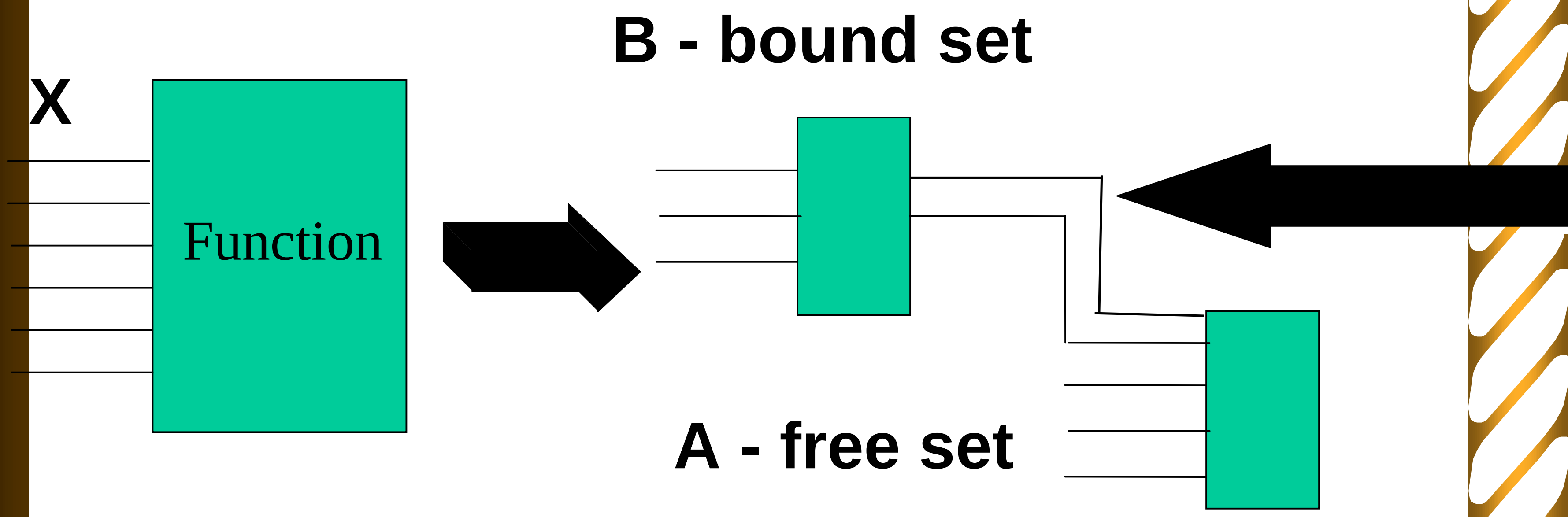


if $A \cap B = \varnothing$, it is *disjoint decomposition*

if $A \cap B \neq \varnothing$, it is *non-disjoint decomposition*

Two-Level Curtis Decomposition

$$F(X) = H(\, G(B), A\,),\quad X = A \cup B$$

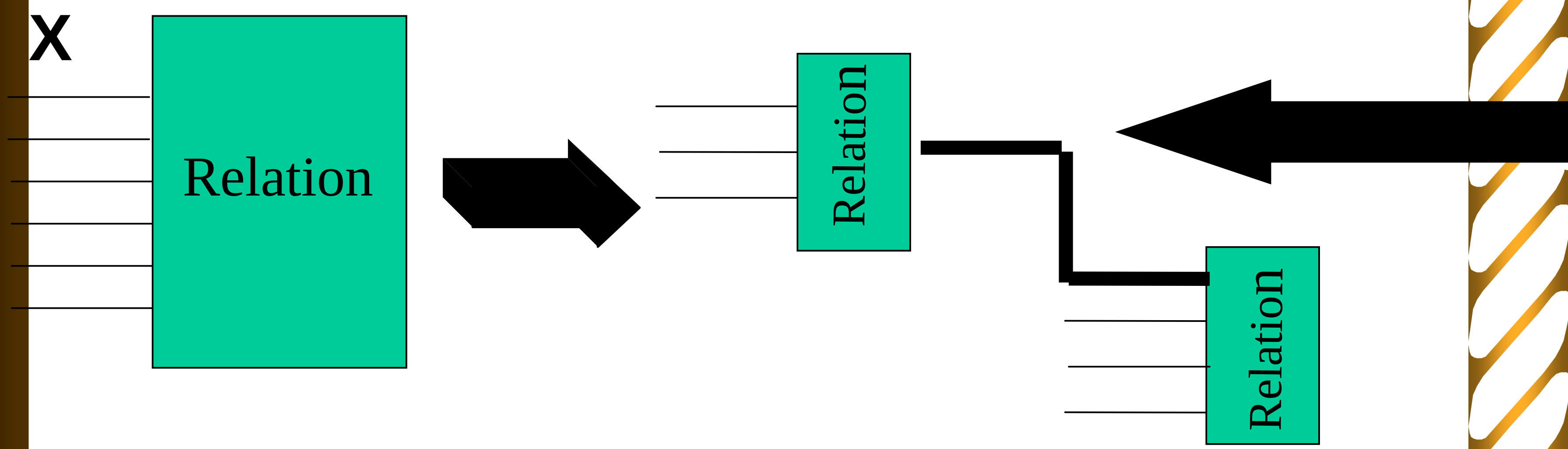


if $A \cap B = \emptyset$, it is *disjoint decomposition*

if $A \cap B \neq \emptyset$, it is *non-disjoint decomposition*

Decomposition of Multi-Valued Relations

$$F(X) = H(\ G(B),\ A\ ),\quad X = A \cup B$$



if $A \cap B = \varnothing$, it is *disjoint decomposition*

if $A \cap B \neq \varnothing$, it is *non-disjoint decomposition*

Applications of Functional Decomposition

- Multi-level FPGA synthesis
- VLSI design
- Machine learning and data mining
- Finite state machine design

Multi-Level Curtis Decomposition

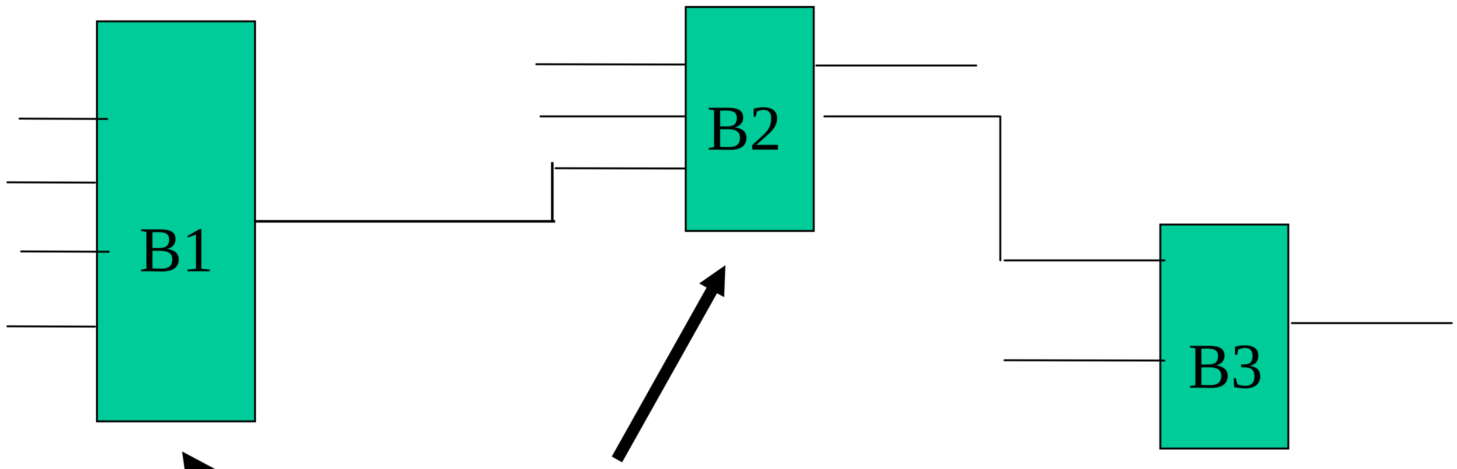
Two-level decomposition is recursively applied to few functions H_i and G_i, until smaller functions G_t and H_t are created, that are not further decomposable.

Thus Curtis decomposition is multi-level, and each two-level stage should create the candidates for the next level that will be as well decomposable as possible.

Cost Function

Decomposed Function Cardinality
is the total cost of all blocks,
where the cost of a binary block with n inputs and m outputs is ***$m * 2^n$.***

Example of DFC calculation



Cost(B1) $=2^4*1=16$

Cost(B2) $=2^3*2=16$

Cost(B3) $=2^2*1=4$

Total DFC = 16 + 16 + 4 = 36

Decomposition Algorithm

- Find a set of partitions (A_i, B_i) of input variables (X) into free variables (A) and bound variables (B)
- For each partitioning, find decomposition $F(X) = H_i(G_i(B_i), A_i)$ such that column multiplicity is minimal, and calculate DFC
- Repeat the process for all partitioning until the decomposition with minimum DFC is found.

Algorithm Requirements

- Since the process is iterative, it is of high importance that minimizing of the column multiplicity index was done as fast as possible.
- At the same time it is important that the value of column multiplicity was close to the absolute minimum value for a given partitioning.

What did we learn?

- Decomposition of Boolean or multi-valued functions and relations can be converted to combinatorial problems that we know.
- Ashenhurst decomposition has been extended by Curtis
- The research is going on to further generalize concepts of Ashenhurst/Curtis decomposition to incompletely specified multi-valued functions, relations, fuzzy logic, reversible logic and other
- Our group has done much research in this area and we continue our research
- Next lecture will show some of possible reductions and method of solving combinational problems

Sources

- Perkowski et al, DAC’99 paper, slides by Alan Mishchenko