Algorithm Implementation

- GPPs (Pentium) - not in this course
- VLSI
  - ASICs - not in this course
  - PLDs
    - PALs/PLAs
      - AND/OR planes
    - Gate Arrays
      - MPGAs
      - FPGAs - focus of this course

PLDs – Programmable Logic Devices

1. AND/OR Arrays (PAL/PLA)

- An array of AND gates connected to an array of OR gates
- SOP form of equations
- PAL
  - AND-plane programmable
  - OR-plane not programmable
- PLA©
  - AND-plane programmable
  - OR-plane programmable

See PLA Example

PLDs cont

2. Gate Arrays

2.A. Mask Programmable Gate Array (MPGA)

- Arrays or rows of uncommitted, prefabricated transistors
  - Wafer is fabbed all the way except for the last metal layers
  - User submits “wire masks” to connect transistors in order to form larger circuits
  - Wafer fab is completed
  - ~ 1-2 months

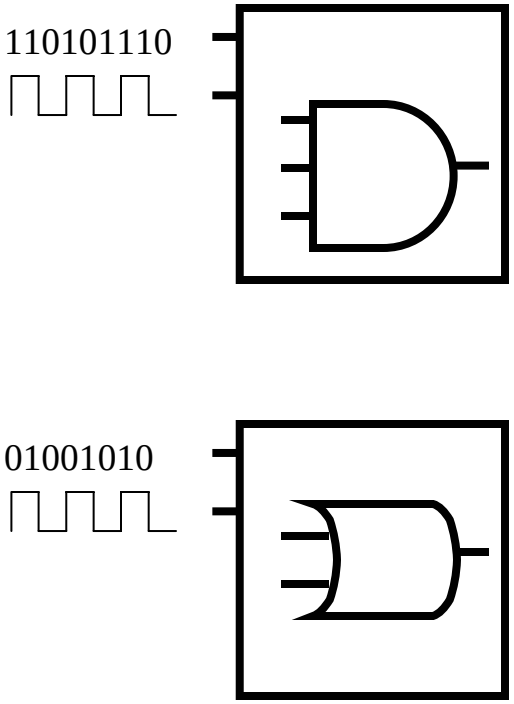
2.B. Field Programmable Gate Arrays (FPGAs)

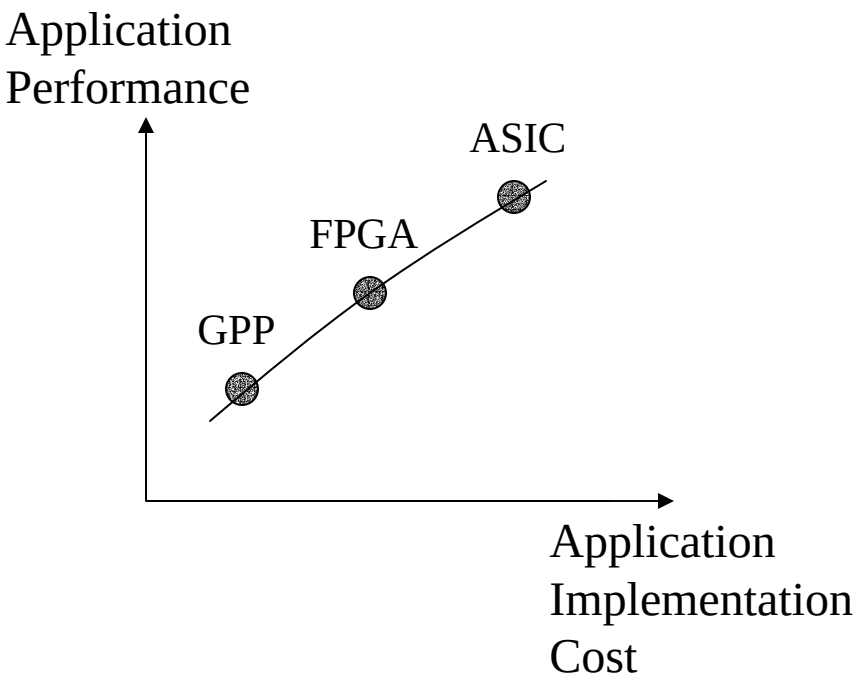
- Pre-fabbed programmable integrated circuit
- Completely user programmed

	FPGA	MPGA
Cost	$	$$
Speed (operation)	Slower	Faster
Implementation	Immediate	Several weeks
Reprogrammability	Yes	No

FPGAs - Definition

- Definition: programmable integrated circuit
- Trade-off between ASICs and general purpose processors (GPP)





FPGAs - Advantages

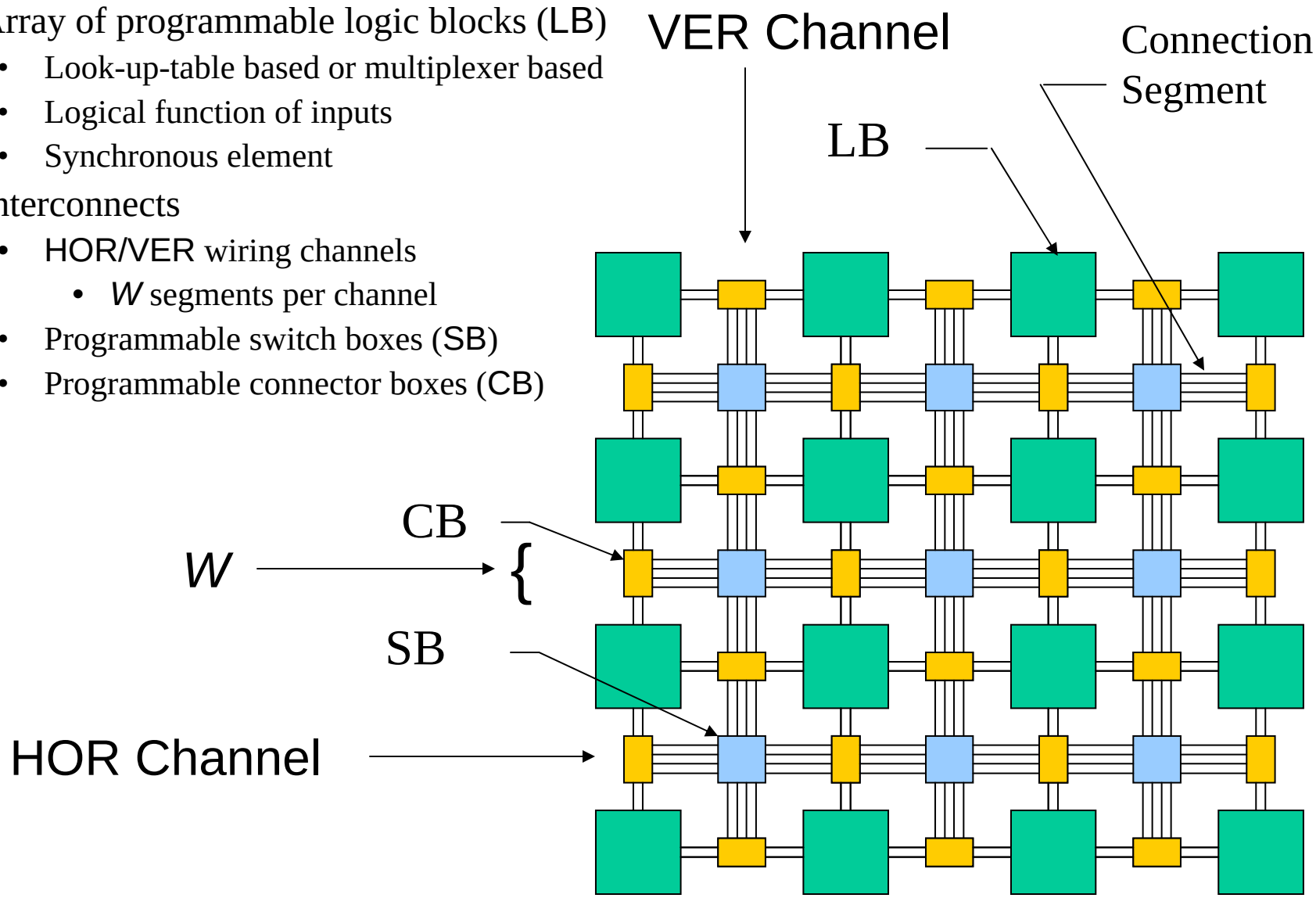
- Reusable
  - ASIC => Fixed design
  - FPGA => Flexible design
- Quick circuit implementation
  - ASIC => ~ 2 months
  - FPGA => ~ 1 day
- Cheap
  - ASIC => More expensive design cycle
    - Verification
  - FPGA => Hardware verification
- Fast execution
  - GPP vs. FPGA

FPGAs - Disadvantages

- Slow application implementation
  - FPGA vs. GPP
- Expensive
  - FPGA vs. GPP
- Slow execution
  - FPGA vs. ASIC
- Area constrained
  - Application size limited
  - Reduce size
  - Split temporally or spatially

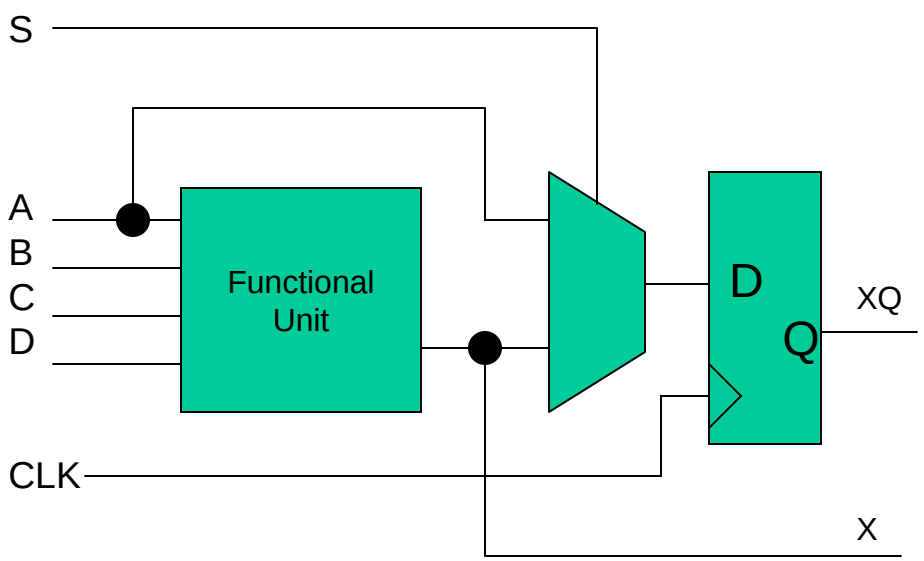
Generic FPGA - Architecture

- Programmable IO (not shown)
- Array of programmable logic blocks (LB)
  - Look-up-table based or multiplexer based
  - Logical function of inputs
  - Synchronous element
- Interconnects
  - HOR/VER wiring channels
    - *W* segments per channel
  - Programmable switch boxes (SB)
  - Programmable connector boxes (CB)



Generic FPGA - Architecture

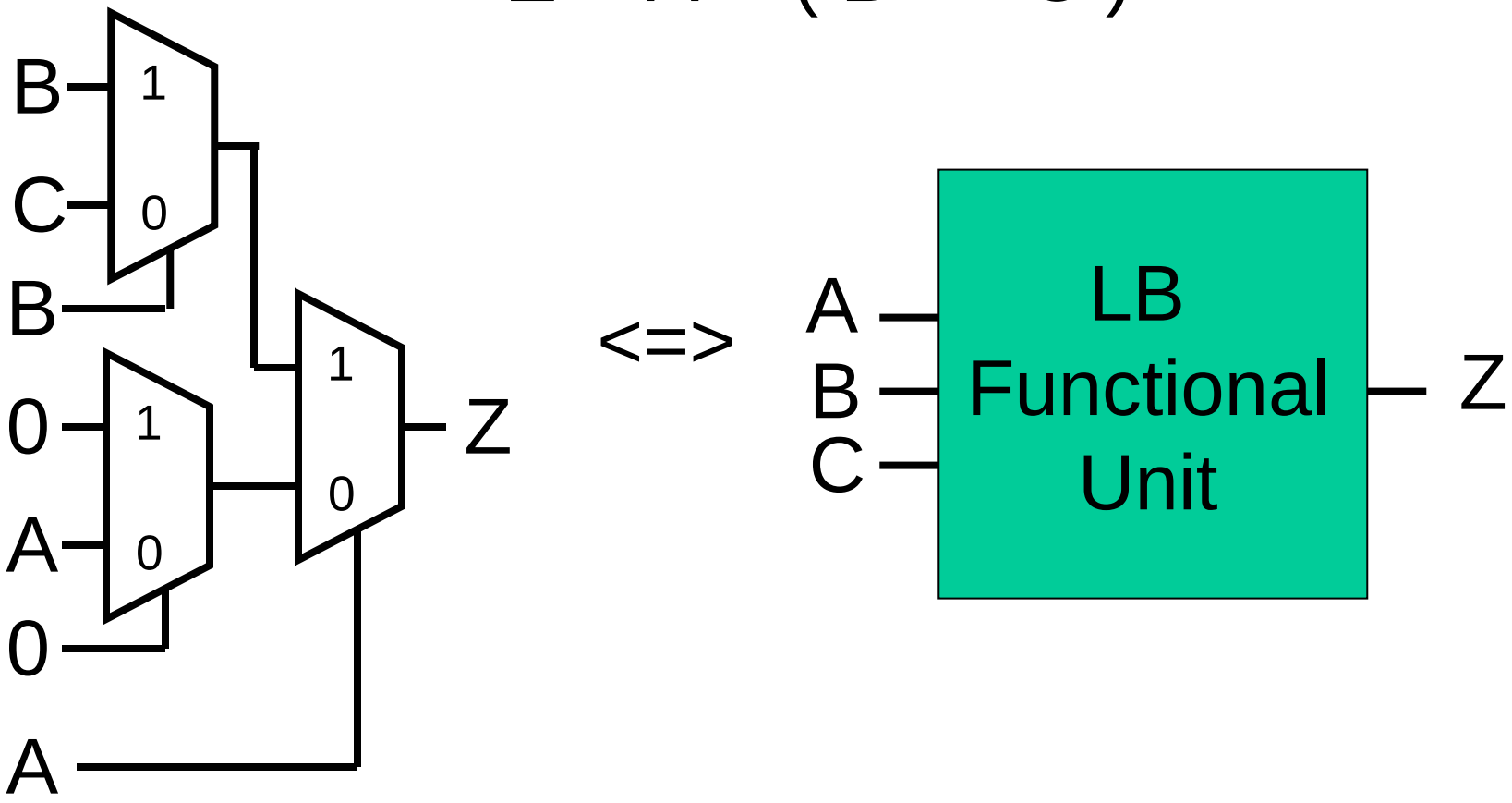
- PLBs or CLBs
  - Functional Unit
    - Nand Gates (PAL like)
    - Multiplexer based
    - LUT based
  - Flip-Flop
  - Usually Multiple Functional/FF units/PLB



Generic FPGA - Architecture

- Logic blocks (LBs) Functional Unit
  - Multiplexer based
  - Look-up-table based (LUT)

$$Z = A \bullet (B + C)$$



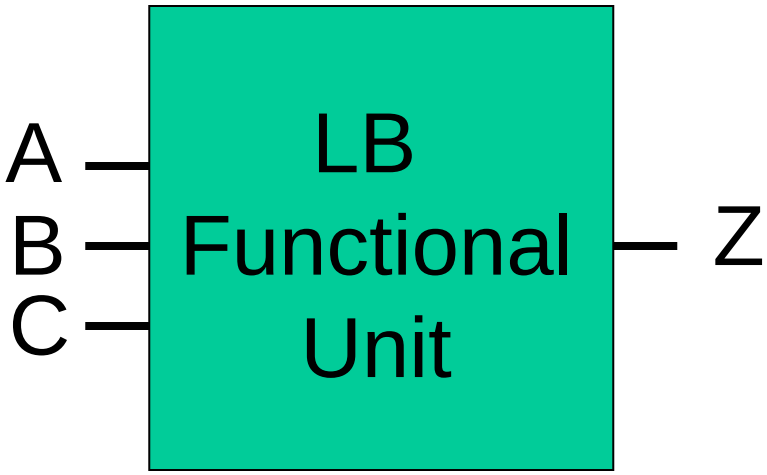
Generic FPGA - Architecture

- Logic blocks (LBs) Functional Unit
  - Multiplexer based
  - Look-up-table based (LUT)

$$Z = A \bullet (B + C)$$

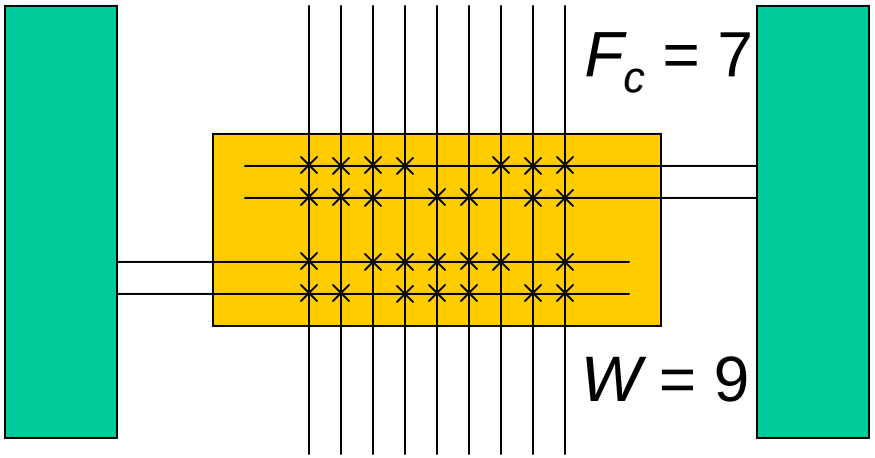
A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

<=>

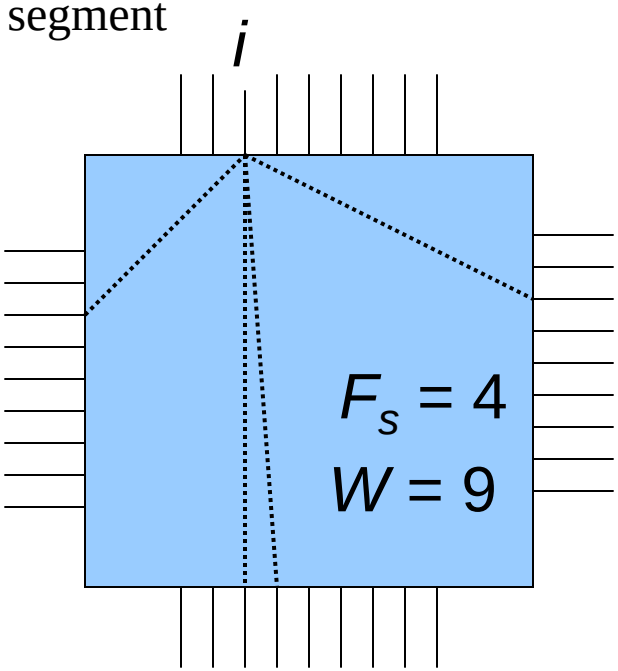


Generic FPGA - Architecture

- Connector Boxes (CB)
  - Connects LB pins to connection segments
  - Flexibility F_C = # possible connections / LB pin
- Switch Boxes (SB)
  - Connects vertical and horizontal wiring channels
  - Flexibility F_S = # possible connections / incoming segment



CB



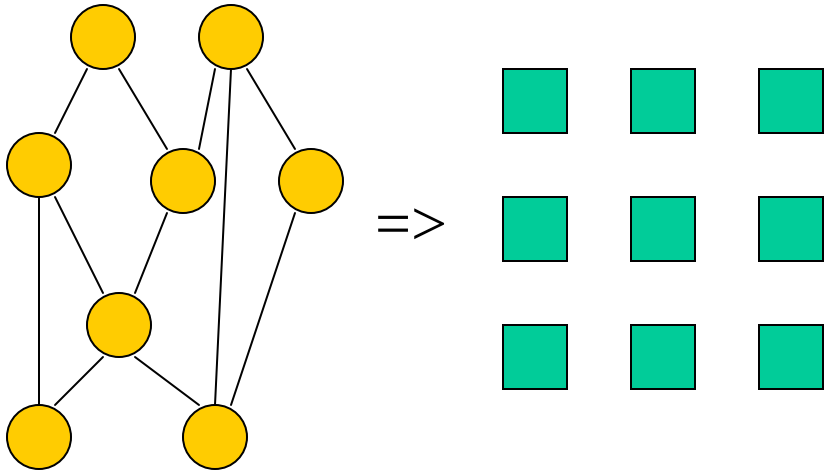
SB

Implementation of FPGA Application

1. Initial Design Entry
2. Translation
3. Logic Optimization and Technology Mapping
4. Placement
5. Routing
6. Configuration File Generation

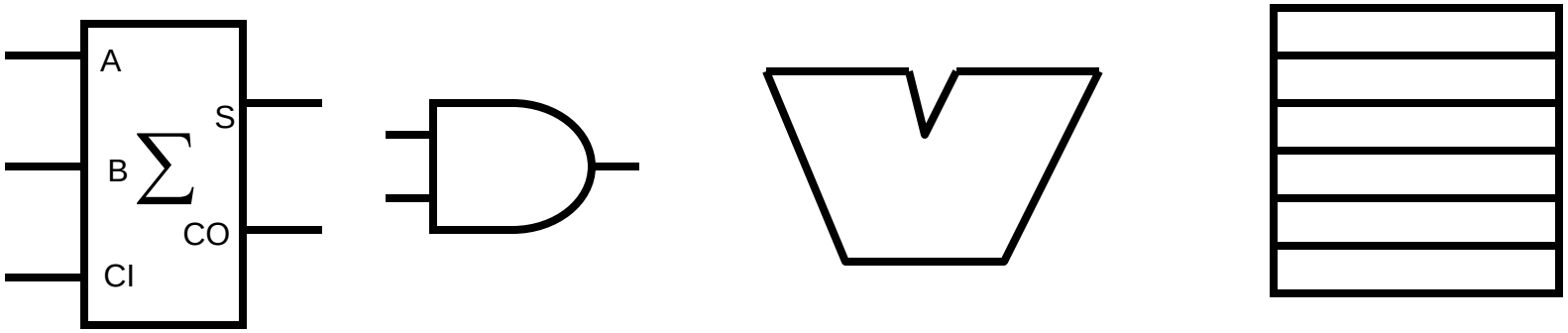
Generic FPGA - Circuit Mapping

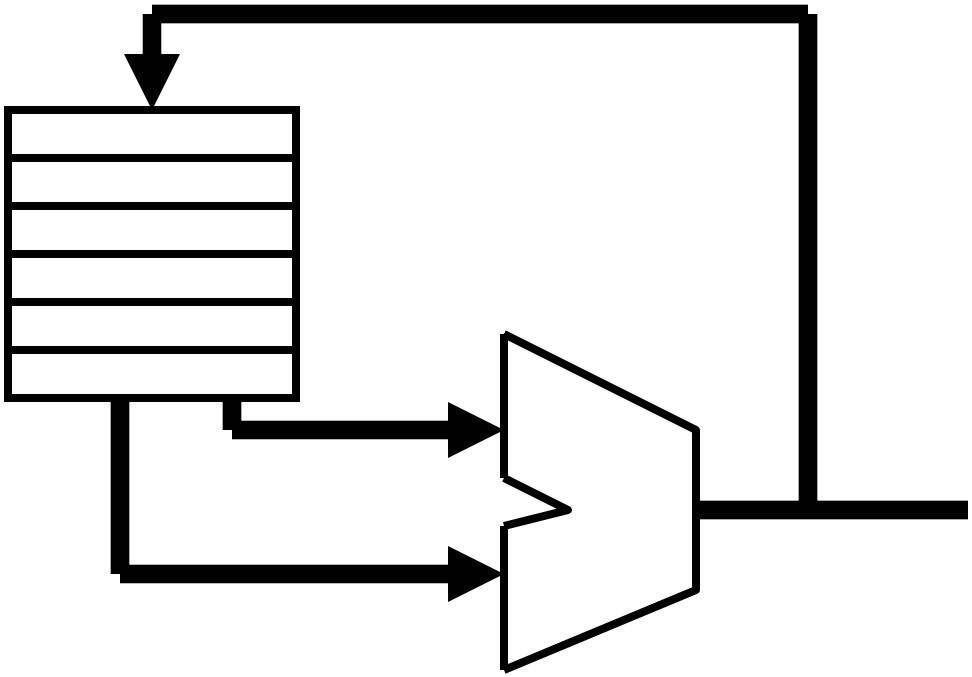
1 Design entry (synthesis,schematic capture …)
2 Technology mapping
3 Placement
4 Routing
5 Bitmap



Generic FPGA - Circuit Mapping

- Design entry
  - Schematic capture





Generic FPGA - Circuit Mapping

- Design entry
  - Synthesis VHDL or Verilog

```
entity ADDER8 is
  port(
      A : in BIT_VECTOR(7 downto 0);
      B : in BIT_VECTOR(7 downto 0);
      S : out BIT_VECTOR(8 downto 0)
  );
end;
architecture BEHAVIORAL of ADDER8 is
 begin
   S <= A + B;
end;
```

•
•
•

```
S(4) =
S(5) = A(5) xor B(5) xor Ci(5)
S(6) = A(6) xor B(6) xor Ci(6)
```

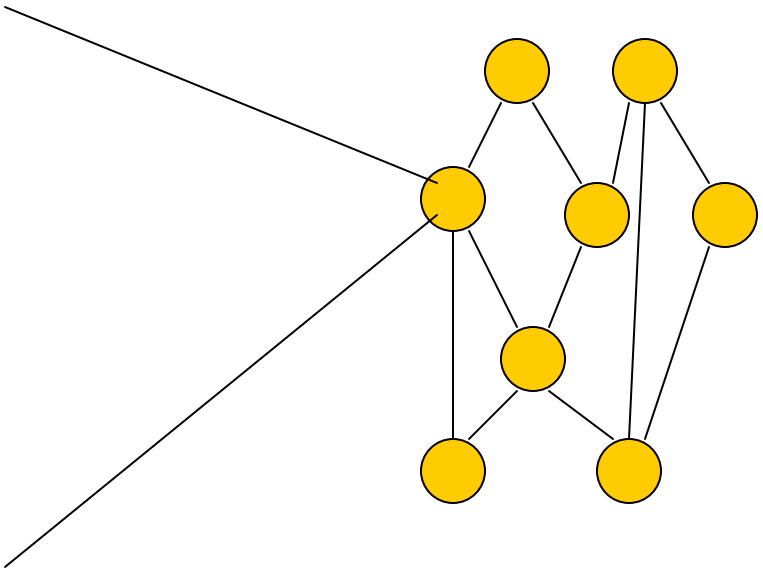
Generic FPGA - Circuit Mapping

- Technology mapping
  - Design equations => logic cell functions and connecting signals

```
S(4) =
S(5) = A(5) xor B(5) xor Ci(5)
S(6) = A(6) xor B(6) xor Ci(6)
```

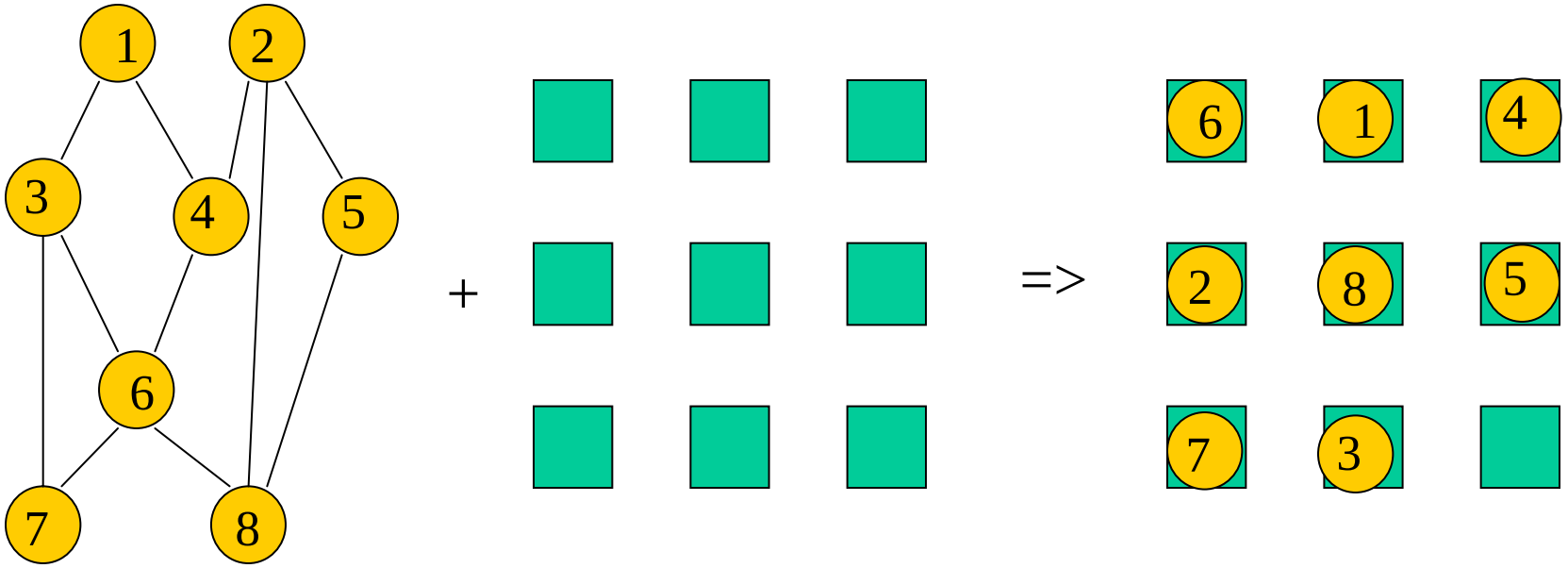
•
•

A(6)	B(6)	Co(6)	S(6)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



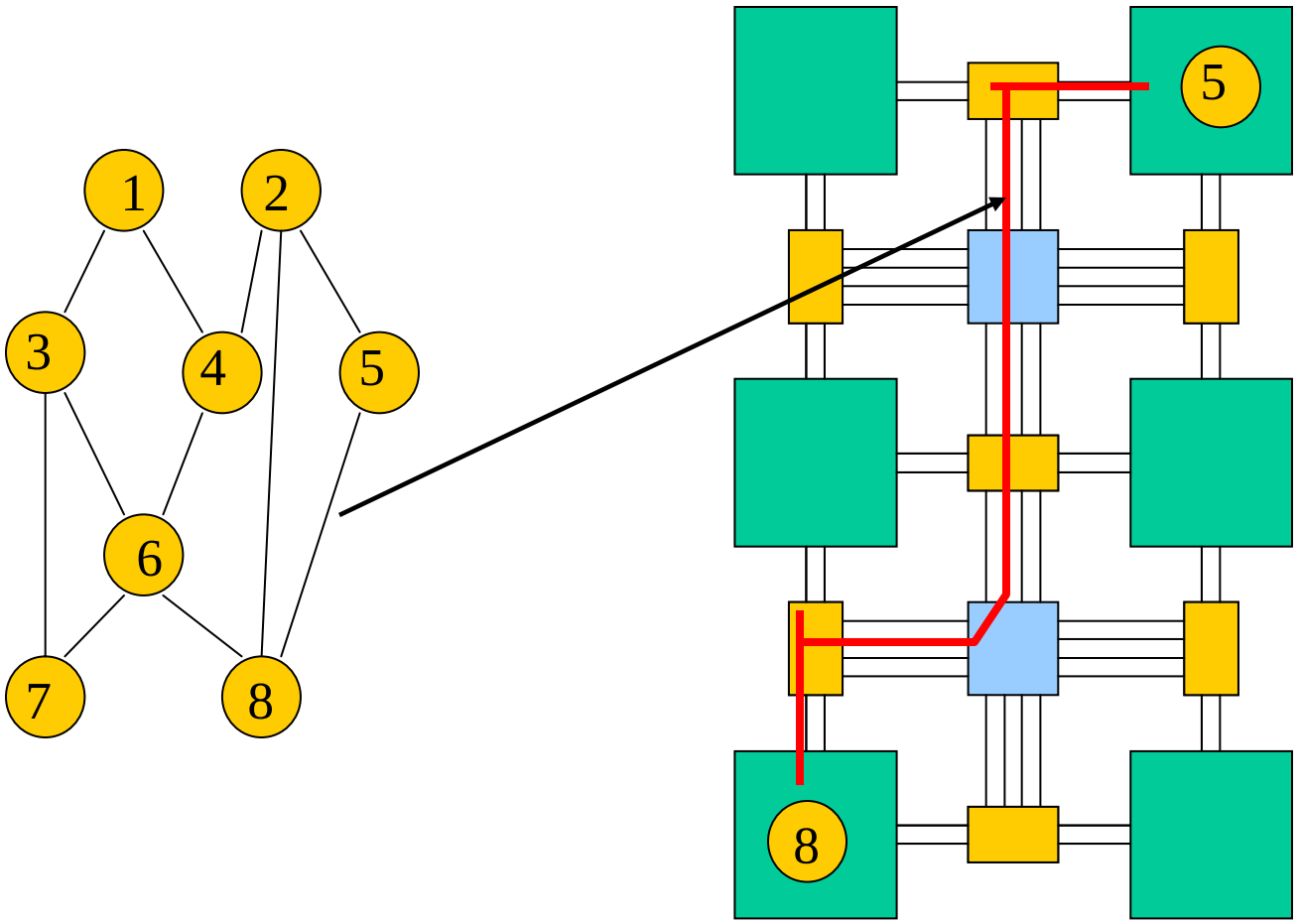
Generic FPGA - Circuit Mapping

- Placement
  - Physical location of logic cell functions on FPGA



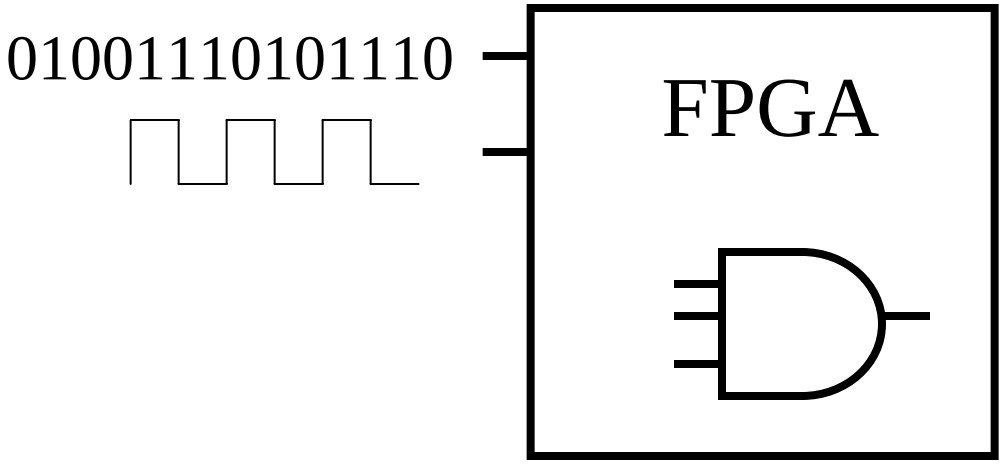
Generic FPGA - Circuit Mapping

- Routing
  - Route connecting signals



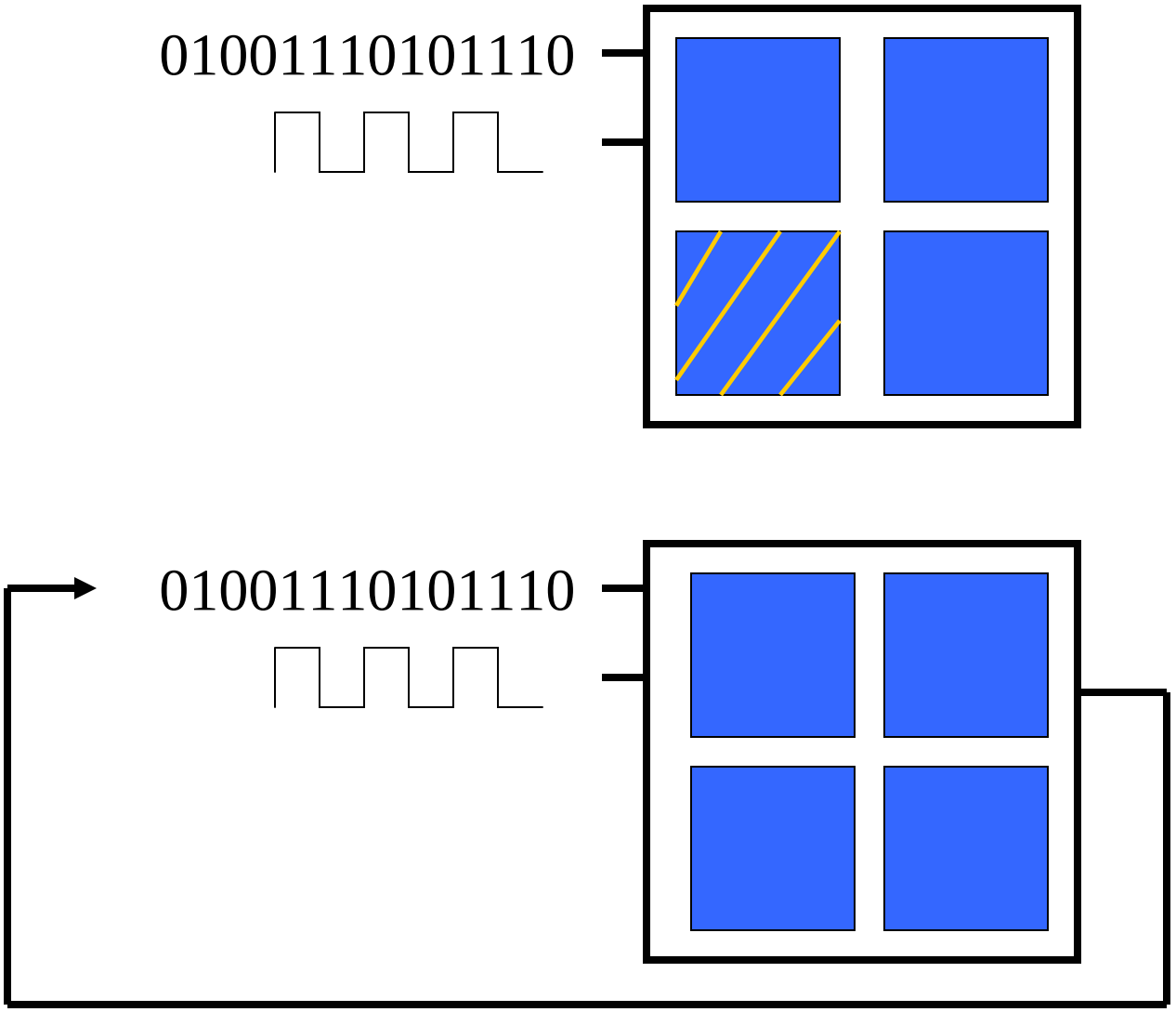
Generic FPGA - Circuit Mapping

- Bitmap
  - Downloadable configuration file
  - Program the FPGA
  - Serial
    - 00100111010010110010100111111011101011•••••
  - Parallel
    - 00100100
    - 00101101
    - ••••



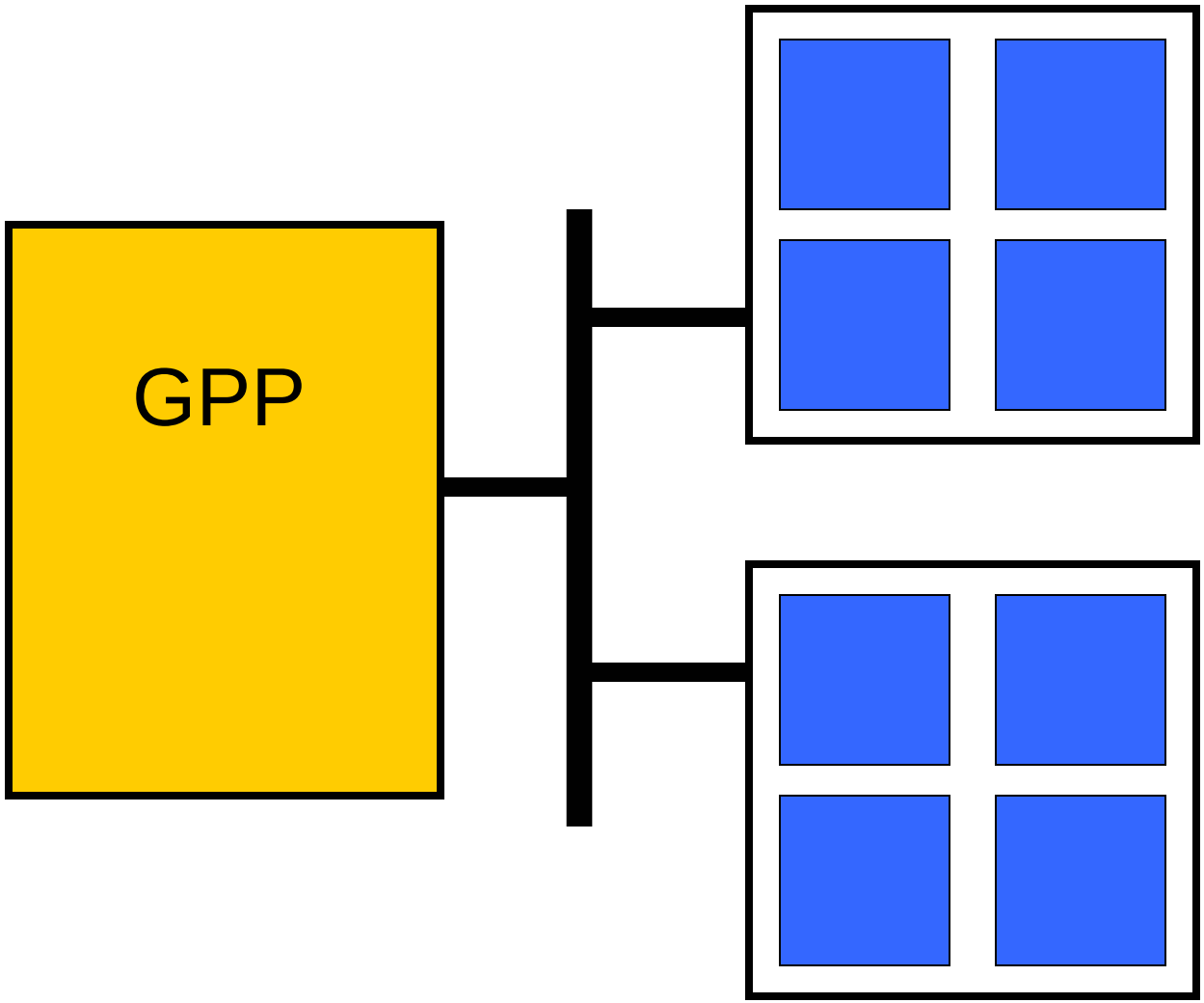
Generic FPGA - Circuit Mapping

- Bitmap
  - Partial
  - Dynamic



Generic FPGA - Circuit Mapping

- GPP with dynamically reconfigurable hardware



Summary

- Introduction
  - Definition
  - Advantages
  - Disadvantages
- Generic architecture
  - Programmable logic blocks
  - Programmable interconnection network
    - Wiring segments
    - Switch boxes
    - Connector boxes
- Circuit mapping process
  - Design entry
  - Technology mapping
  - Placement
  - Routing
  - Configuration
- Summary