DIGITAL LOGIC

Chapter 7
Memory

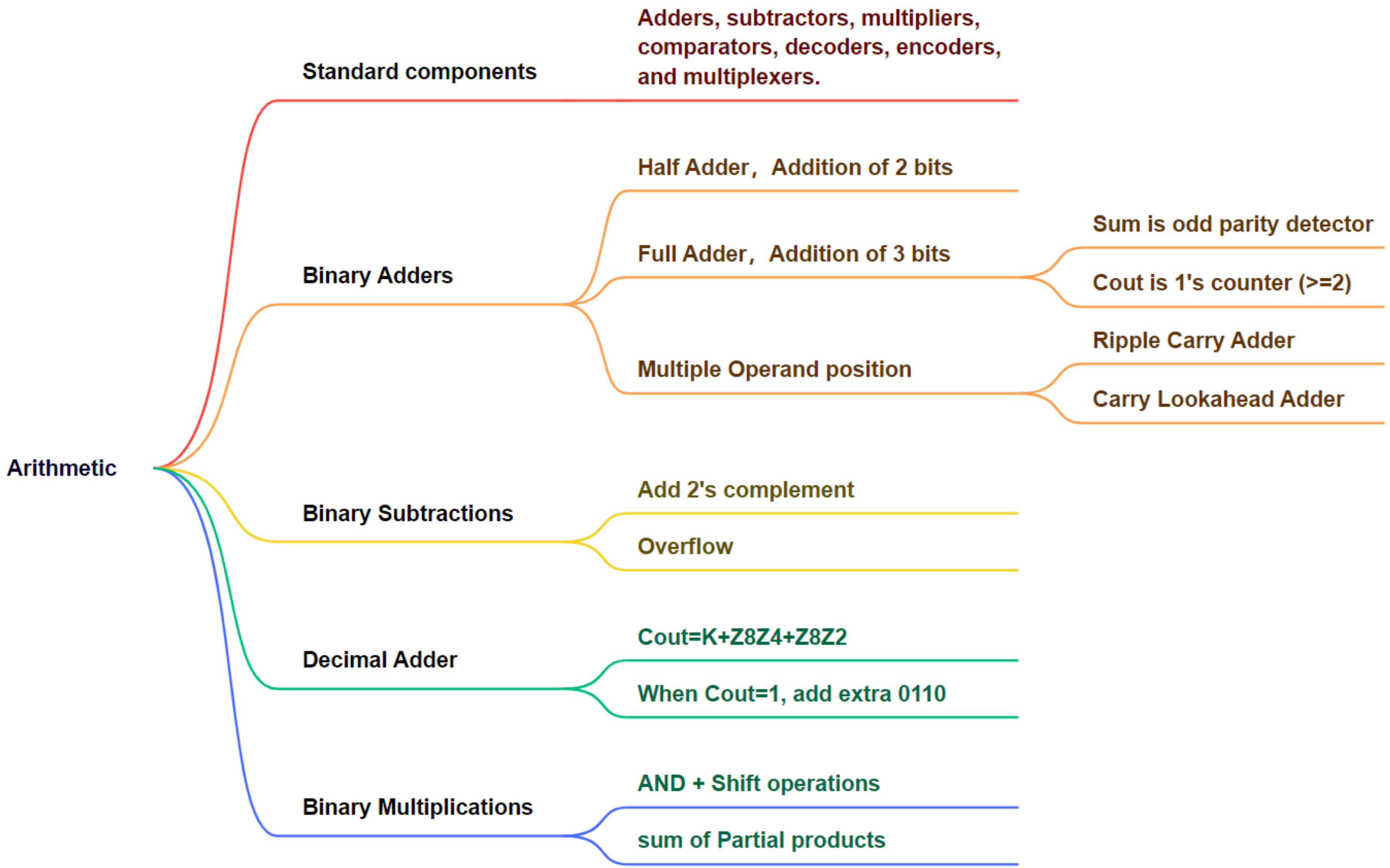
2024 Fall

baiyh@sustech.edu.cn

Today's Agenda

- Recap
  - Asynchronous Counter
  - Synchronous Counter
    - Binary counter
    - Ring counter
    - Johnson counter
  - Design a synchronous counter
- Context
  - RAM
  - ROM
  - PLA
  - FPGA
- Reading: Textbook, Chapter 7

Recap


Arithmetic
Standard components
Adders, subtractors, multipliers, comparators, decoders, encoders, and multiplexers.
Binary Adders
Half Adder，Addition of 2 bits
Full Adder，Addition of 3 bits
Sum is odd parity detector
Cout is 1's counter (>=2)
Multiple Operand position
Ripple Carry Adder
Carry Lookahead Adder
Binary Subtractions
Add 2's complement
Overflow
Decimal Adder
Cout=K+Z8Z4+Z8Z2
When Cout=1, add extra 0110
Binary Multiplications
AND + Shift operations
sum of Partial products

Outline

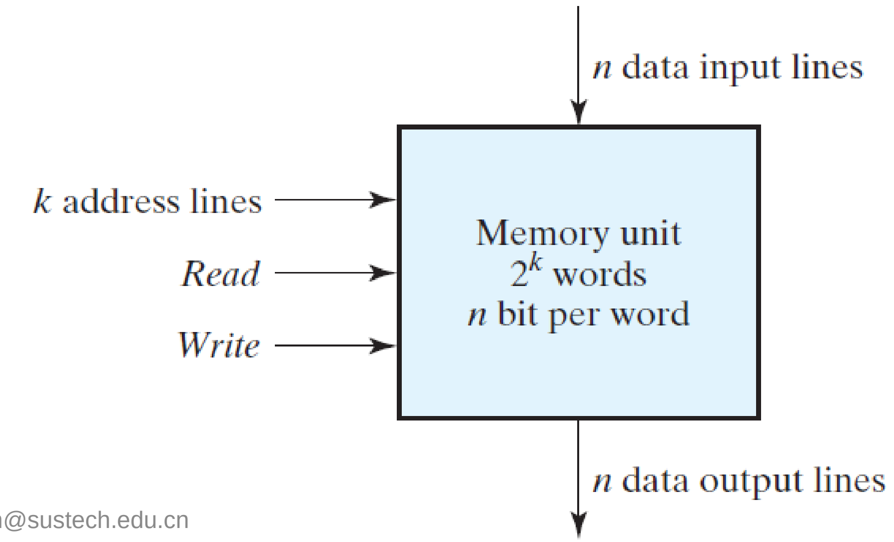
- Random-Access Memory
- Memory Decoding
- Read-Only Memory
- Programmable Logic Array
- Sequential Programmable Devices-FPGA

Memory Unit

- A collection of storage cells together with associated circuits needed to transfer information in and out of storage
- RAM: Random-Access Memory
  - Volatile (memory units that lose stored information when power is turned off)
  - To accept new information for storage to be available later for use
  - read/write operation
- ROM: Read-Only Memory
  - Nonvolatile
  - The information inside can not be altered by writing
  - Programmable devices are specified by some hardware procedure

Random-Access Memory

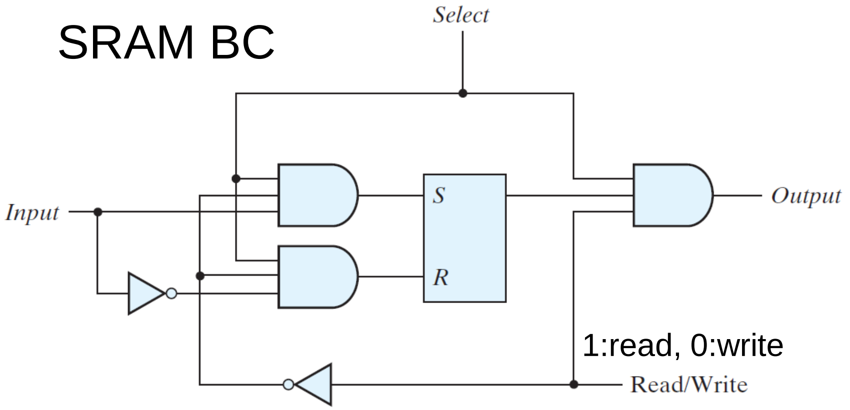
- Characteristics
  - The time it takes to transfer data to or from any desired location is always the same.
  - A memory unit stored binary information in groups of bits (words)
  - The size of the RAM is 2^k x n bits. It has k address lines, n input data lines and n output data lines.
  - For a commodity RAM, n=8, 16, 32 or 64.
- The communication between a memory and its environment is achieved through
  - Data input lines
  - Data output lines
  - Address selection lines
  - Control lines (read and write)



Memory Content Array & Memory Cell

- A memory with k-bit address has 2^k words (depth)
  - (depth)x(width) memory
  - eg. 1K x 16 memory (1024x16 memory)
- A memory cell is also called a binary storage cell (BC)

SRAM BC



Memory address		Memory content
Binary	Decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022 2^k-2	0000110100011110
1111111111	1023 2^k-1	1101111000100101
k-bit address		n bits data

Static RAM vs. Dynamic RAM

- A memory cell (MC) can be considered as a clocked D latch with an AND gate and an output driver
- For a **static RAM (SRAM),** MC is constructed by 6 transistors
  - using cross-coupled inverters to serve as a latch
  - and implementing the input AND gate and the output driver with one transistor each.
- For a **dynamic RAM (DRAM)**, MC is constructed by only 1 transistor
  - The latch is implemented by a capacitor.
  - It needs to be refreshed periodically (read and write back).
  - It has high density (therefore low cost)

Write and Read Operation

- Write operation
  - Apply the binary address to the address lines
  - Apply the data bits to the data input lines
  - Activate the write input (0)
- Read operation
  - Apply the binary address to the address lines
  - Activate the read input (1)

Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Timing Waveforms

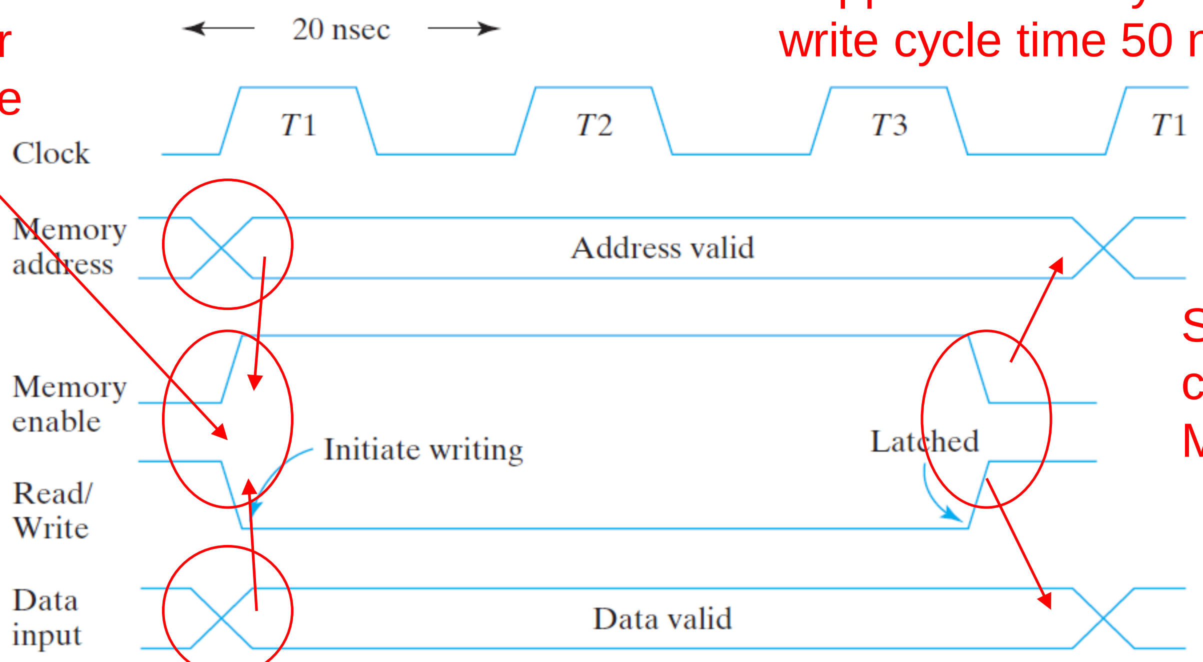
- The operation of the memory unit is controlled by an external device such as a CPU
- The access time is the time required to select a word and read it
- The write cycle time is the time required to complete a write operation
- Read and write operations must be controlled by CPU and be synchronized with an external clock.

Timing Waveforms

- A write cycle
  - T1: providing the address and input data to the memory, and the write/read control signal
  - Address bus and read/write control must stay active for 50ns
  - The address and data signal must remain stable for a short time after control signal is deactivated.

Suppose memory access time and write cycle time 50 ns

activated after address stable

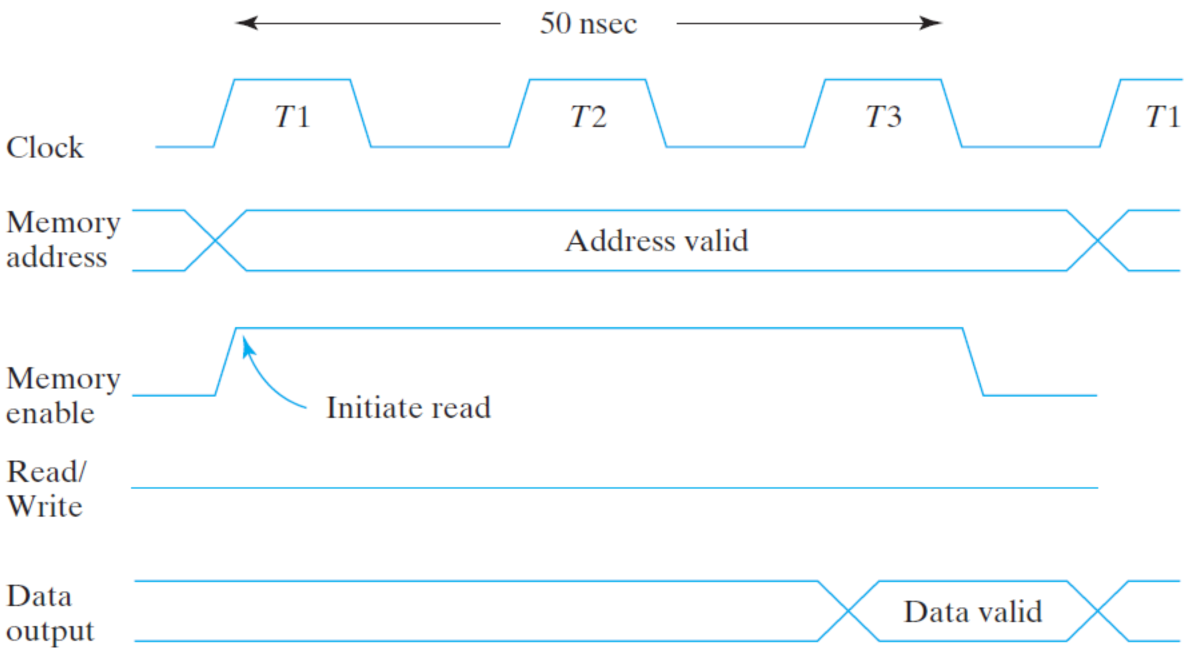


(a) Write cycle

Suppose CPU clock frequency 50 MHz 20ns period)

Timing Waveforms

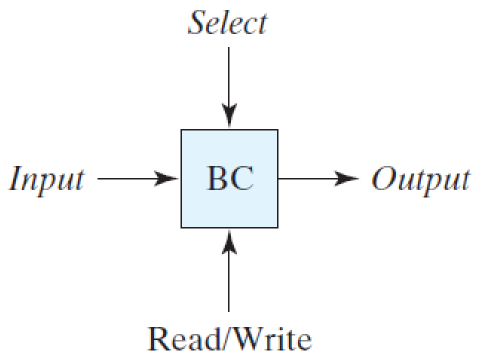
- A read cycle

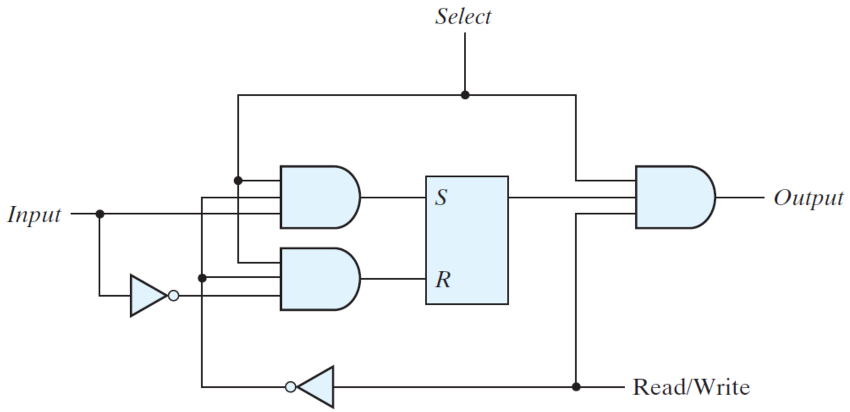


(b) Read cycle

Memory Unit and Internal Construction

- A memory unit has two parts
  - The storage components (memory cell)
  - The decoding circuits to select the memory word
- A RAM of m (2^k) words and n bits per word
  - m*n binary storage cells
  - Decoding circuits to select individual words
    - k-to-2^k decoder: address input to word line
    - Read/Write control
    - Input data/Output data

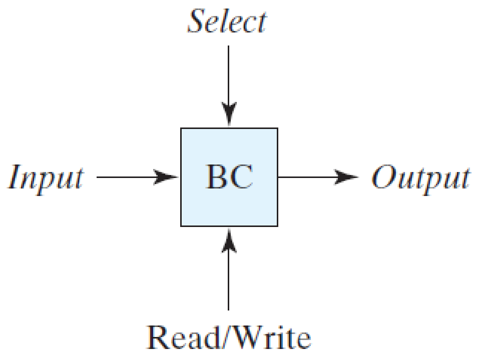


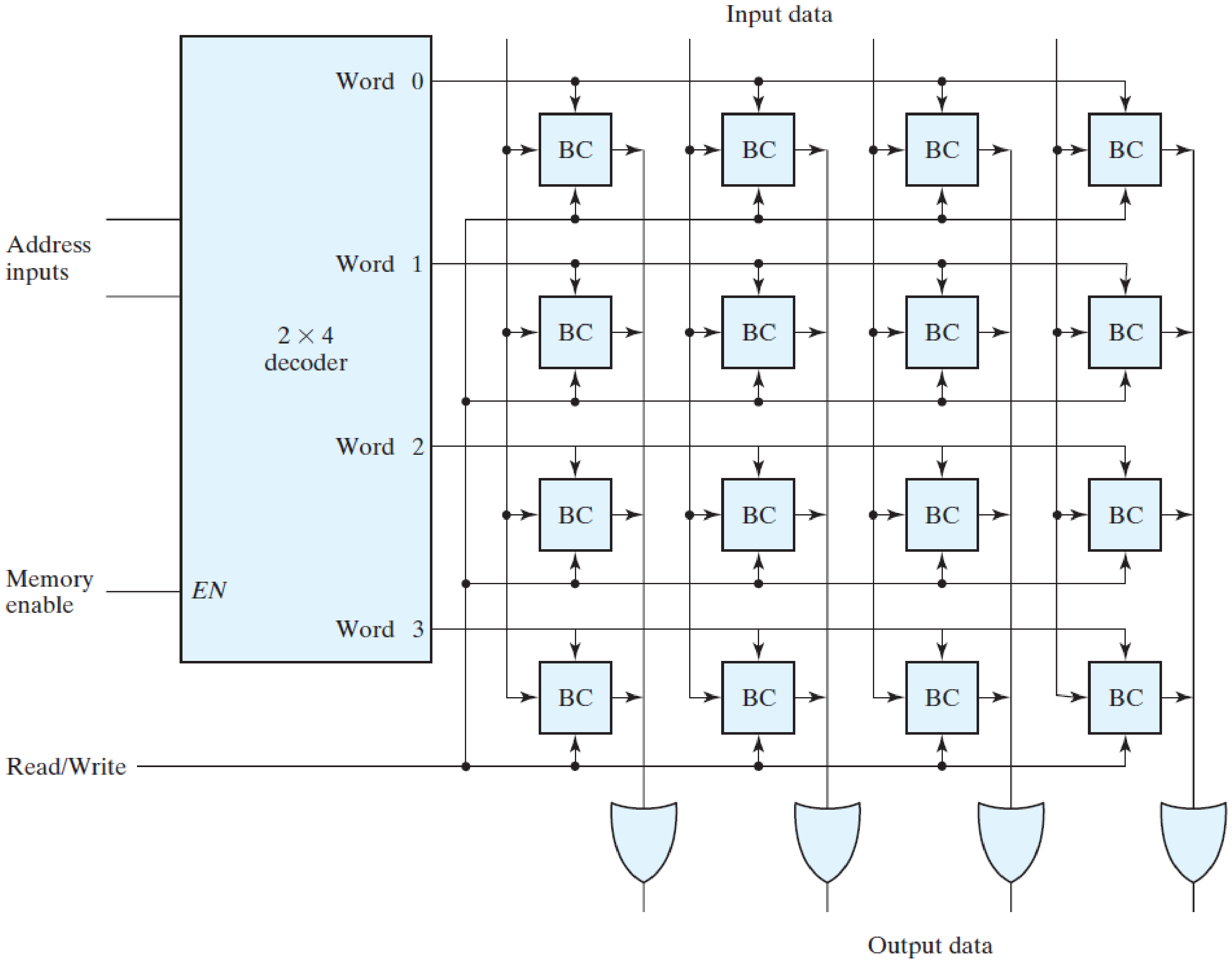


memory cell (BC)

4x4 RAM

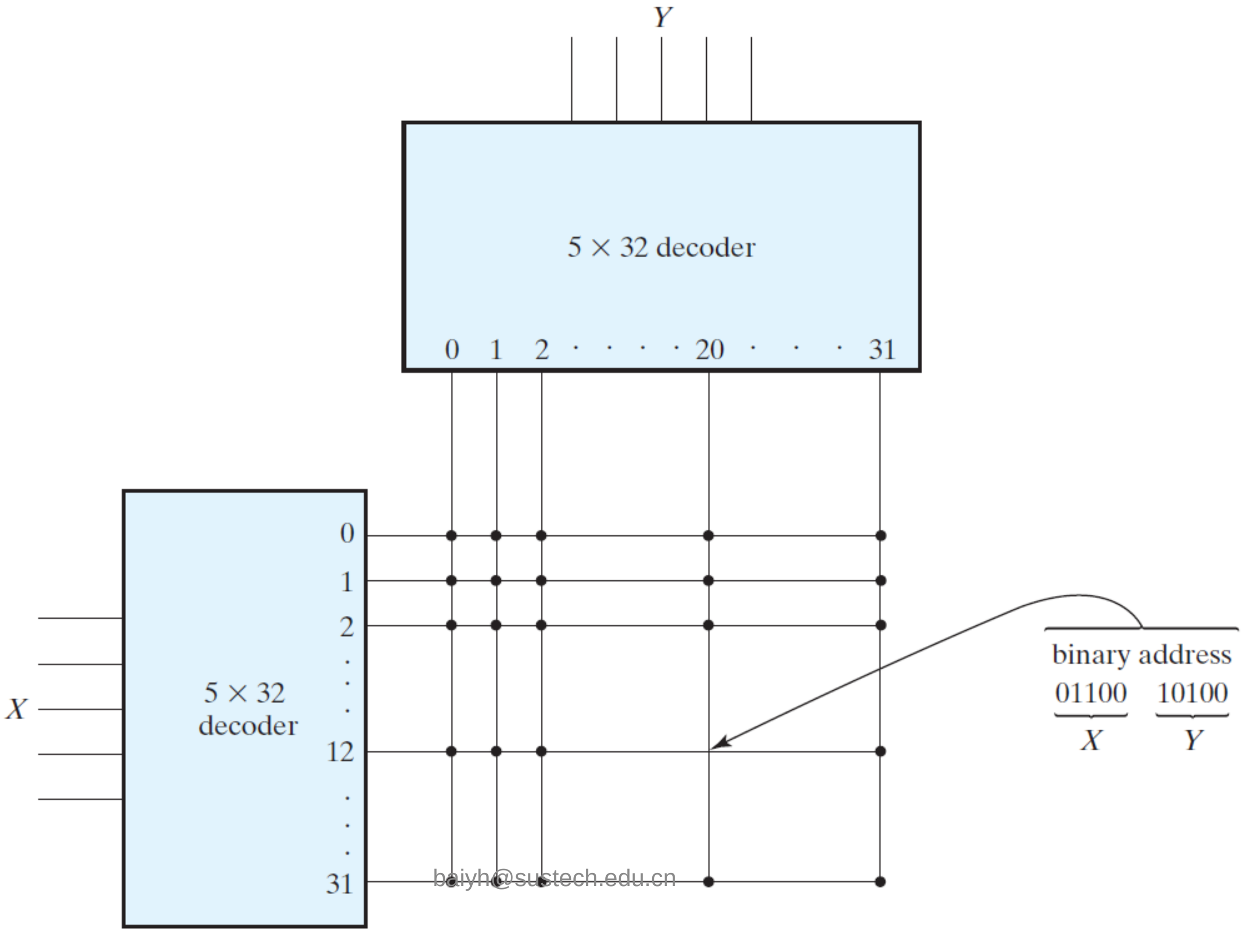
check the decoder
with enable input
in Chapter 4





Coincident Decoding

- A two-dimensional selection scheme
  - To reduce the complexity of the decoding circuits
  - To arrange the memory cells in an array that is close as possible to square

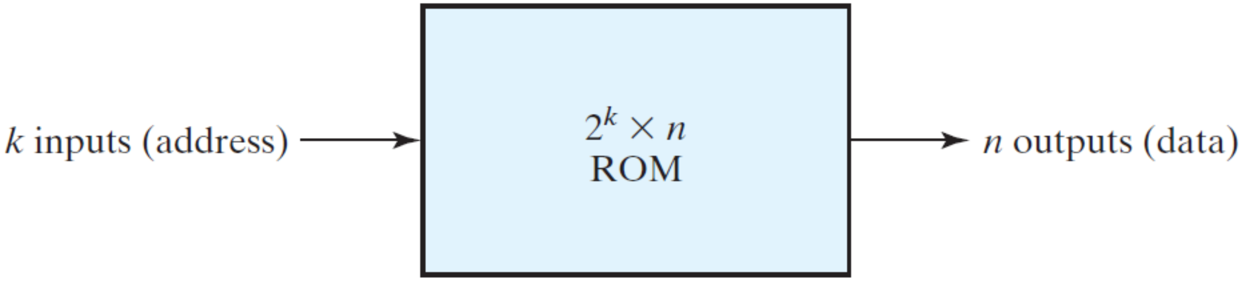


Coincident Decoding

- Example: 1k-word memory
  - A 10-to-1024 decoder
    - 1024 AND gates with 10 inputs per gate
  - Two 5-to-32 decoders
    - 2*(32 AND gates with 5 inputs per gate)
    - Each word in the memory is selected by the coincidence between 1 of 32 rows and 1 of 32 columns for a total 1024 words
    - Two dimensional decoding structure reduces the circuit complexity and the cycle time of the memory

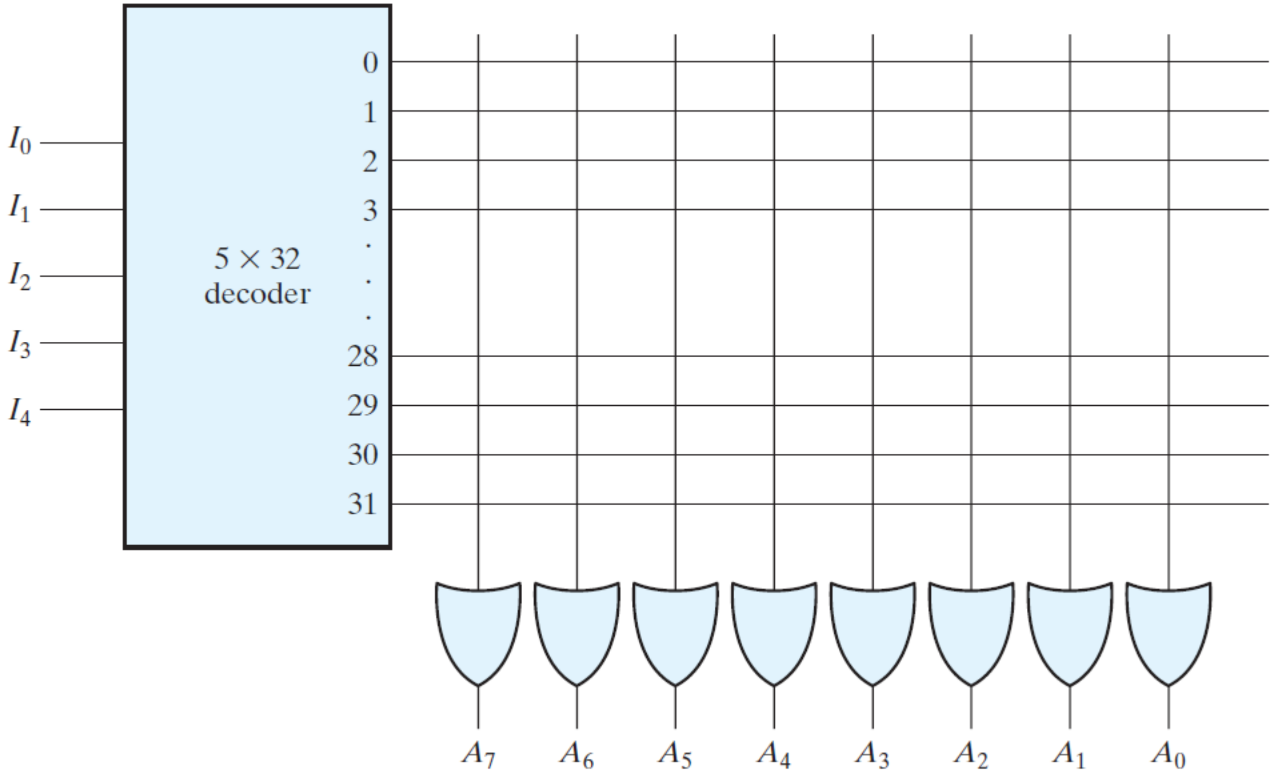
Read-Only Memory (ROM)

- ROM stores permanent binary information
- Once the pattern is established in the ROM, it stays within the unit, no matter power is on or off
- 2^k x n ROM
  - k address input lines
  - enable input(s)



32x8 ROM

- 32 x 8 ROM (2^5 x 8)
  - 5-to-32 decoder (k=5)
  - 32(2^5) outputs of the decoder are connected to each of the eight OR gates, which have 32(2^5) inputs
  - 32 x 8 internal programmable connections



ROM Programming

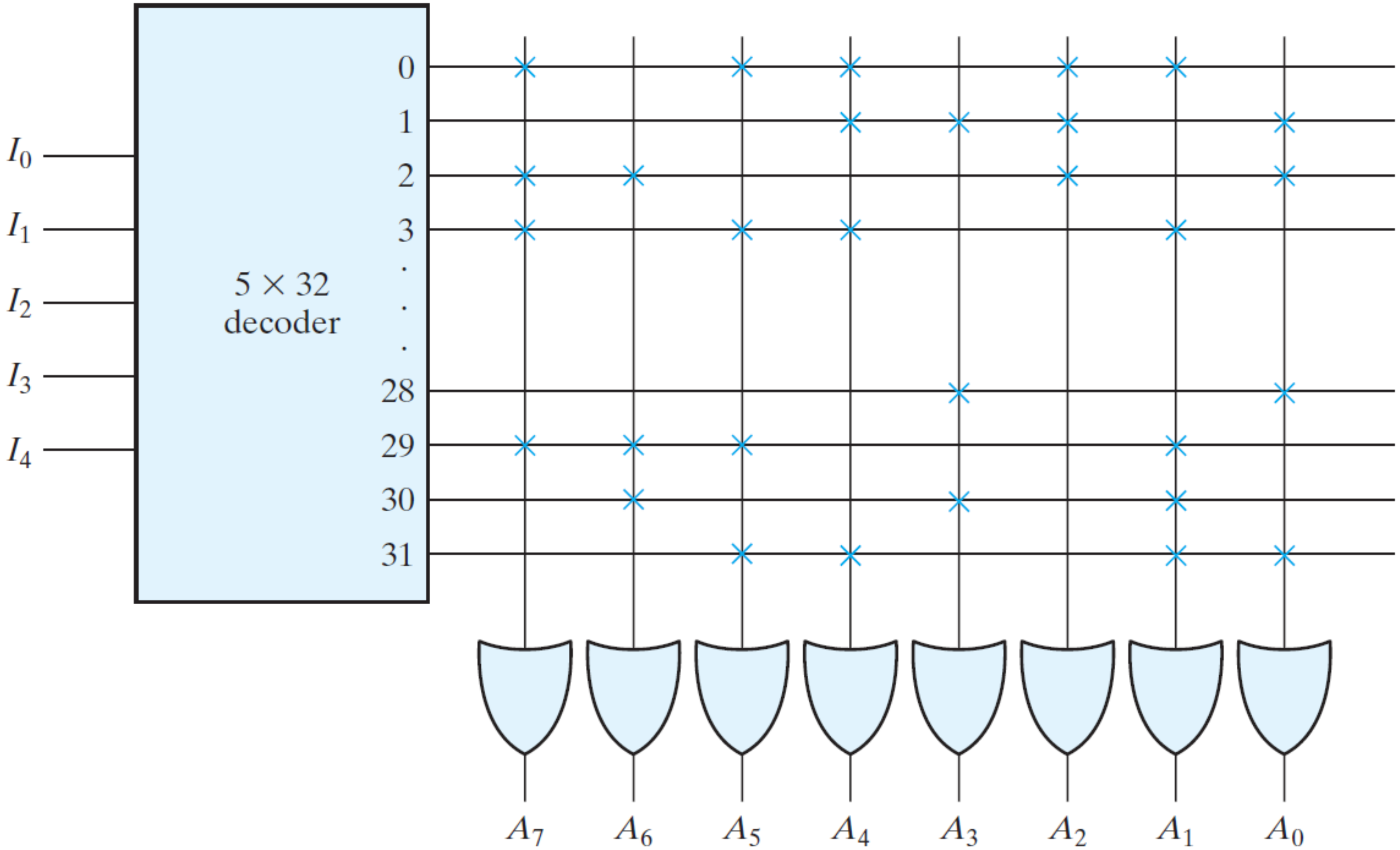
- PROM(Programmable ROM) with programmable interconnections
- close [1]: connected to high voltage
- open [0]: ground left
- A fuse that can be blown by applying a high voltage pulse

ROM Truth Table (Partial)

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮							⋮			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

ROM Programming

- ROM programming according to ROM table using fusing



Combinational Circuit Implementation

- The internal operation of ROM can be interpreted in two ways
  - A memory contains a fixed pattern of stored words
  - A unit that implements a combinational circuit as sum of minterms.
- ROM: a decoder + OR gates
  - a Boolean function = sum of minterms
  - Ex. $A_7(I_4,I_3,I_2,I_1,I_0) = \sum(0,2,3,\ldots,29)$
  - For an n-input, m-output combinational circuit
  - 2^n x m ROM
- Design Procedure
  - Determine the size of ROM
  - Obtain the programming truth table of ROM
  - The truth table = the fuse pattern

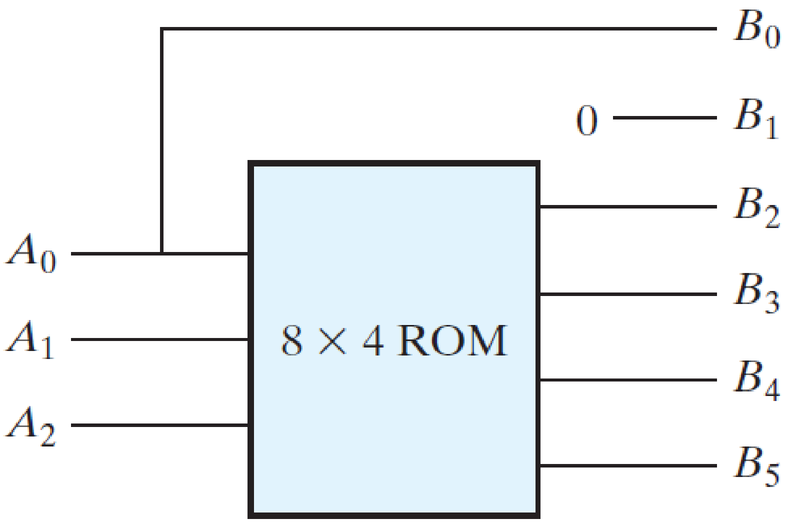
ROM Implementation

- A combinational circuit has 3 inputs, 6 outputs and the truth table, construct the circuit

Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

$B_0=A_0$, $B_1=0$, leave only 8x4 ROM

ROM Implementation



(a) Block diagram

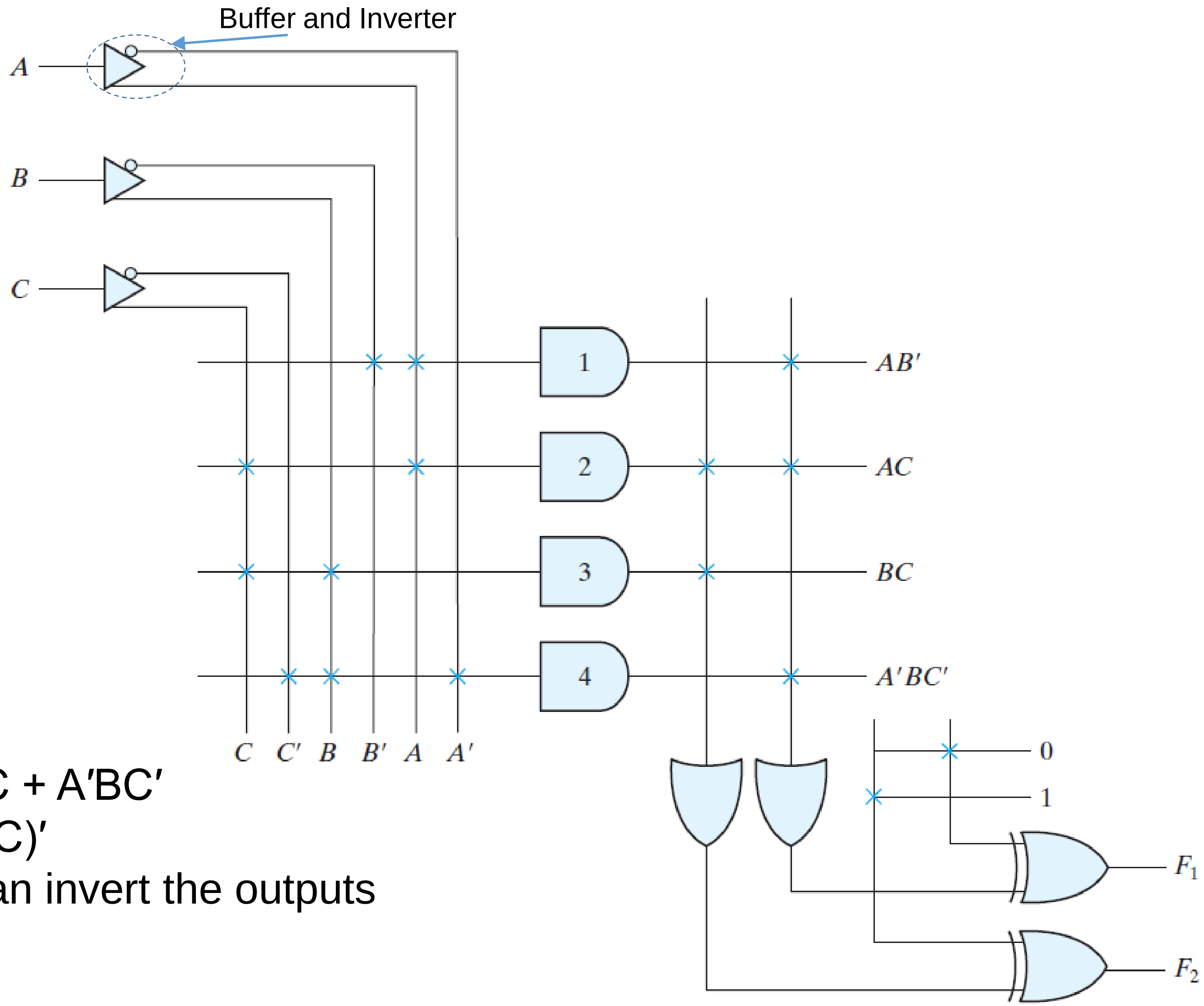
A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM truth table

Programmable Logic Array

- PLA has two main parts
  - An array of programmable AND gates
    - Can generate any product terms of the inputs
  - An array of programmable OR gates
    - Can generate the sums of the products
- PLA is more flexible than PROM
  - PLA uses any combination of products while PROM uses sum of minterms
- PLA uses less circuits than PROM
  - Only the needed product terms are generated in PLA while all minterms must be generated in PROM

PLA



F1 = AB′ + AC + A′BC′
F2 = (AC + BC)′
XOR gates can invert the outputs

PLA

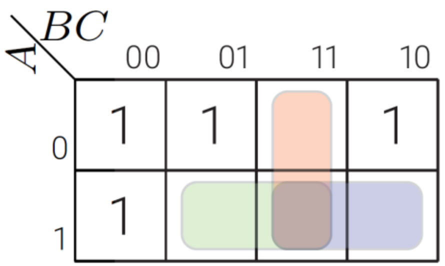
- Specify the fuse map
  - 1st col: list the product terms numerically
  - 2nd col: specify the requires paths between inputs and AND gates
  - 3rd col: specify the required paths between AND gates and OR gates
  - (T)(C) stand for true or complement for programming XOR

PLA Programming Table

		Inputs			Outputs (T)	Outputs (C)
	Product Term	A	B	C	F_1	F_2
AB'	1	1	0	—	1	—
AC	2	1	—	1	1	1
BC	3	—	1	1	—	1
$A'BC'$	4	0	1	0	1	—

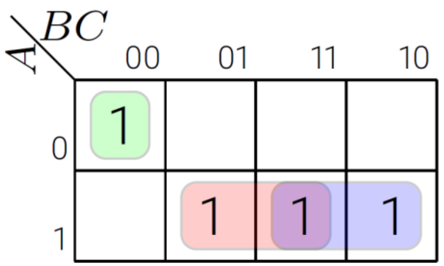
Example

- Implement the following two Boolean functions with a PLA:
- $F_1(A, B, C) = \Sigma\ (0, 1, 2, 4)$
- $F_2(A, B, C) = \Sigma\ (0, 5, 6, 7)$
  - Both true and complement of the function should be simplified to check



$F_1' = AB + AC + BC$

$F_1 = (AB + AC + BC)'$



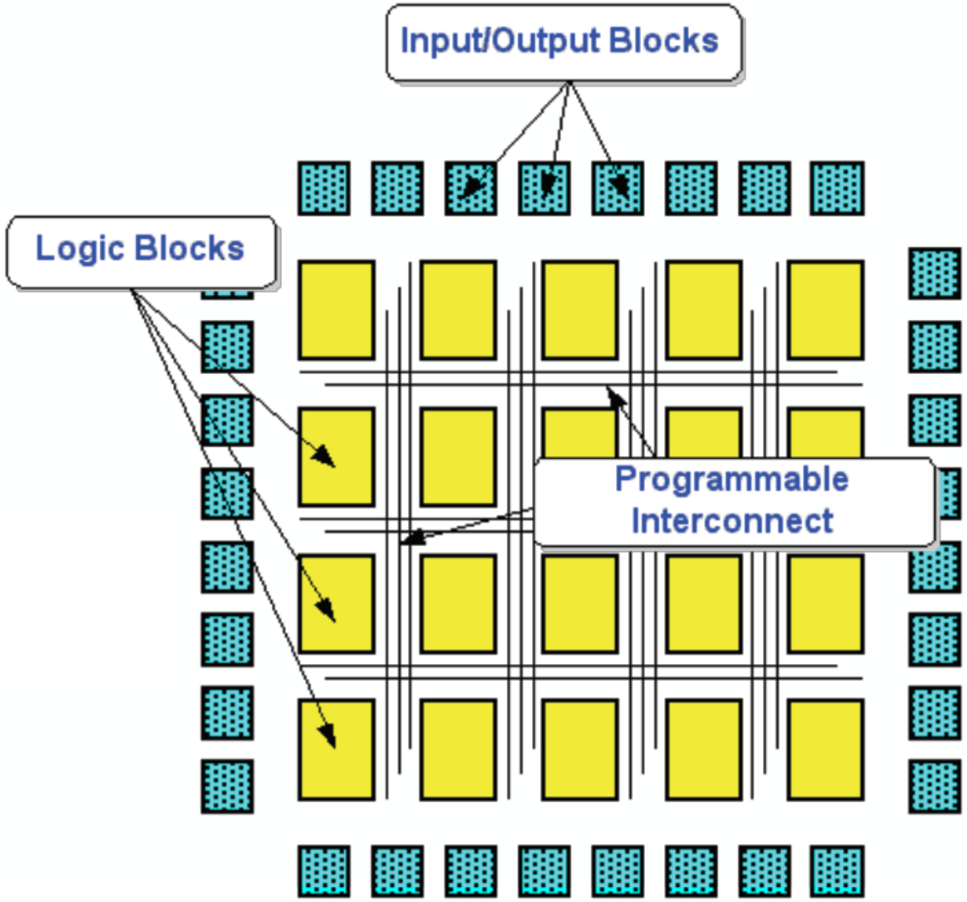
$F_2 = AB + AC + A'B'C'$

PLA programming table

Product term		Inputs A	B	C	Outputs (C) F_1	(T) F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1

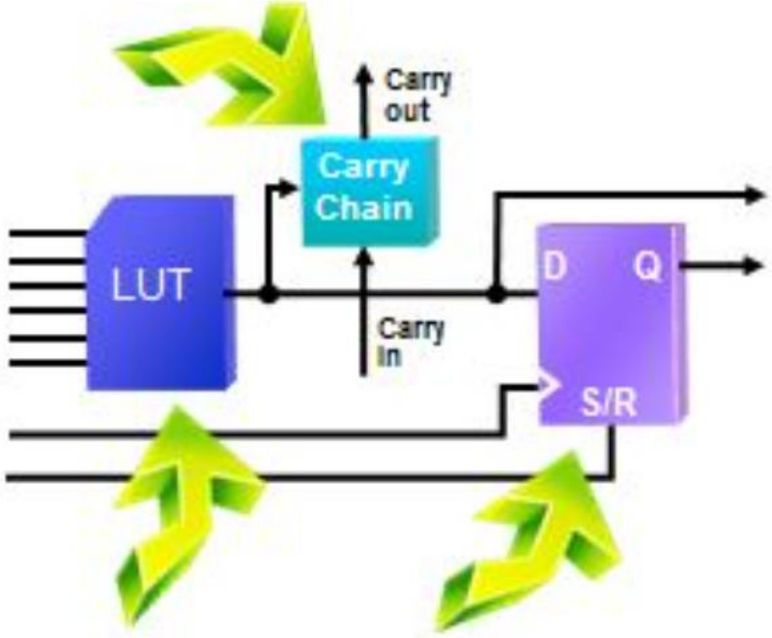
FPGA

- What is in the array? All sorts of stuff…
  - I/O Cells
  - Logic Cells
  - Memories
  - Microprocessors
  - Clock Management
  - High Speed I/O Transceivers
  - Programmable routing

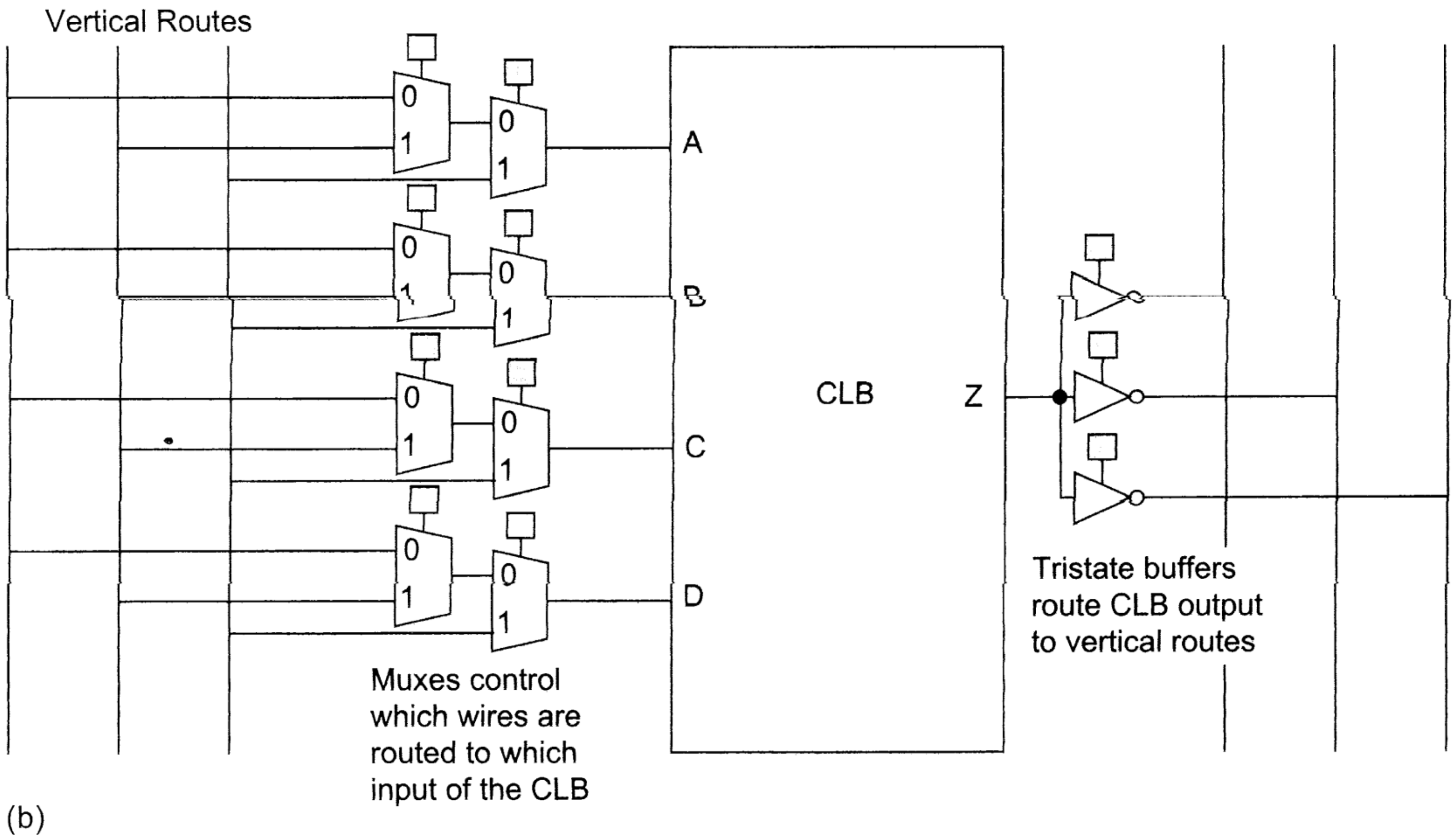


Simple FPGA Logic Cell

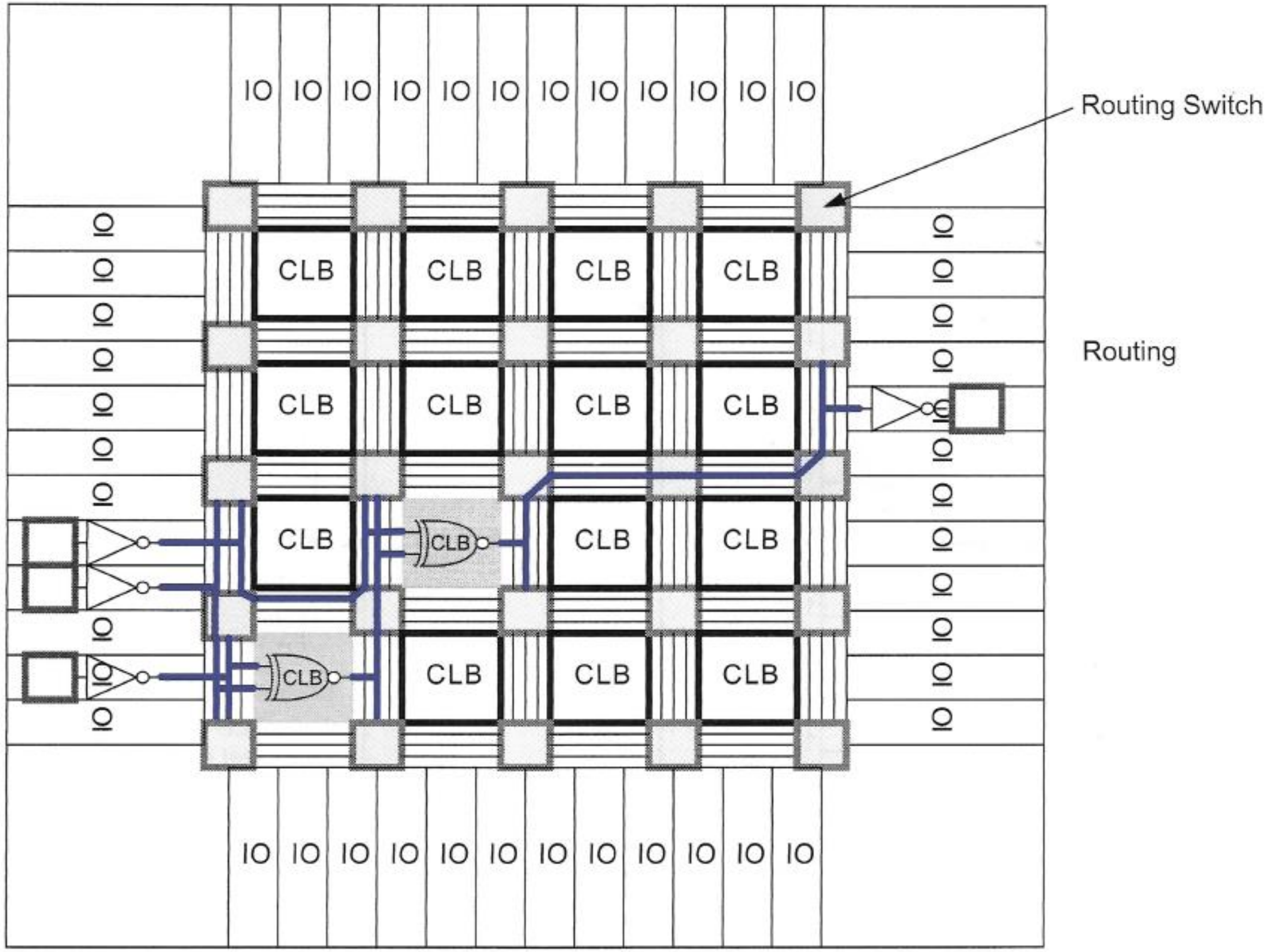
- Logic cells include
  - Combinatorial logic, arithmetic logic, and a reg
- Combinatorial logic is implemented using
- Register functions can include latches, JI s
- Arithmetic logic is a dedicated carry chair netic operations



Simple FPGA Routing Cell



Routed FPGA



IO IO IO IO IO IO IO IO IO IO IO IO
Routing Switch
Routing
CLB CLB CLB CLB
CLB CLB CLB CLB
CLB CLB CLB CLB
CLB CLB CLB CLB
IO IO IO IO IO IO IO IO IO IO IO IO

JKFF FPGA implementation

Characteristic table

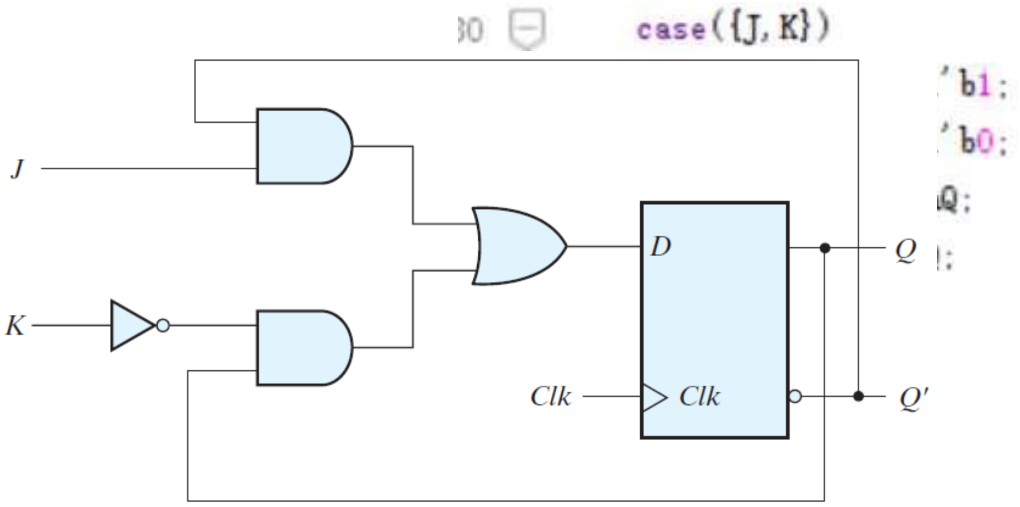
***JK* Flip-Flop**			
J	***K***	***Q*(*t* + 1)**	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

Derive the Truth table

J	K	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Characteristic equation of JKFF

$$\boxed{Q(t+1) = JQ(t)' + K'Q(t)}$$



JKFF FPGA implementation