


Chapter 7: Digital Components

Prof. Ben Lee
Oregon State University
School of Electrical Engineering and Computer Science

1



Chapter Goals

- Review basic digital design concepts:
 - Designing basic digital components using logic gates and memory elements:
 - Decoders/encoders, multiplexers, counters, registers, memories, and Arithmetic and Logic Units (ALUs).
- Understand that these digital components represent fundamental building blocks for any digital system, especially processor (microarchitecture) design:
 - Modular design.

Chapter 7: Digital Components 2

2

Contents

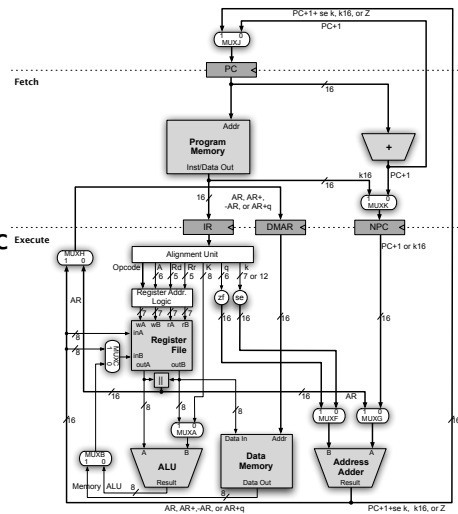
- 7.1 Introduction
- 7.2 Multiplexers
- 7.3 Decoders
- 7.4 Memory Elements
- 7.5 Registers
- 7.6 Register File
- Basic ALU (see Chapter 9)

7.1 Introduction

Basic Digital Components

- Multiplexors
- Decoders/encoders
- Registers
- Memories
- ...later we will see Arithmetic and Logic Units (ALUs)

AVR Microarchitecture



Chapter 7: Digital Components

5

5

7.2 Multiplexer

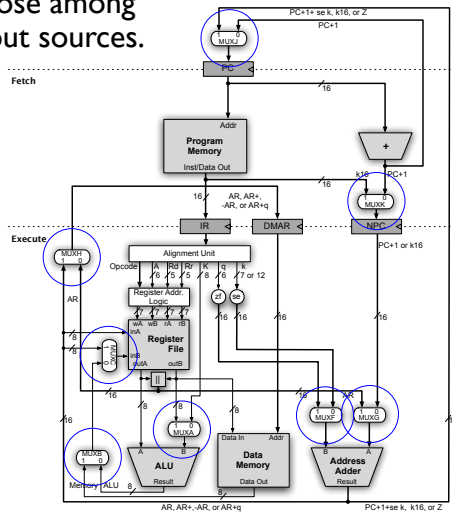
Chapter 7: Digital Components

6

6

Multiplexors

Use to choose among multiple input sources.



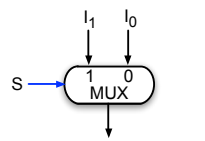
Chapter 7: Digital Components

7

7

Multiplexors

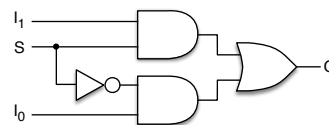
- 2^n data inputs, n control inputs, one output.
- Control signal pattern forms binary index of input connected to output.



I_1	I_0	S	O
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$I_1 \backslash I_0$	00	01	11	10
0		1	1	
1			1	1

$$O = S' I_0 + S I_1$$



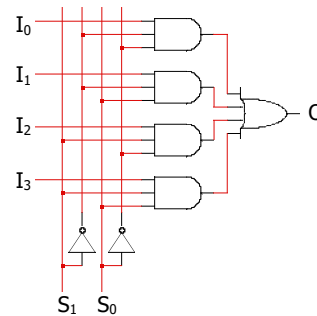
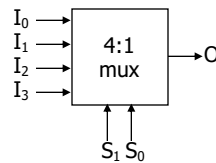
Chapter 7: Digital Components

8

8

Multiplexors (cont.)

- 2:1 MUX: $O = S'I_0 + SI_1$
- 4:1 MUX: $O = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$
- 8:1 MUX: $O = S_2'S_1'S_0'I_0 + S_2'S_1'S_0I_1 + S_2'S_1S_0'I_2 + S_2'S_1S_0I_3 + S_2S_1'S_0'I_4 + S_2S_1'S_0I_5 + S_2S_1S_0'I_6 + S_2S_1S_0I_7$



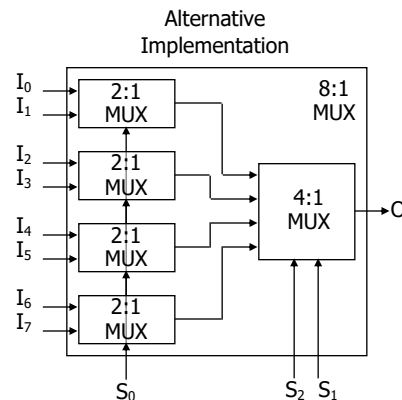
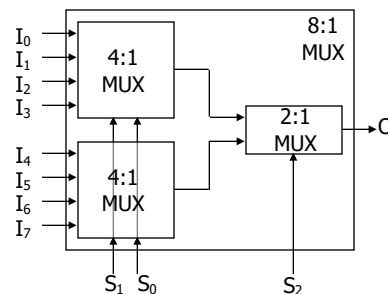
Chapter 7: Digital Components

9

9

Cascading Multiplexors

- Large multiplexers implemented by cascading smaller ones.



Chapter 7: Digital Components

10

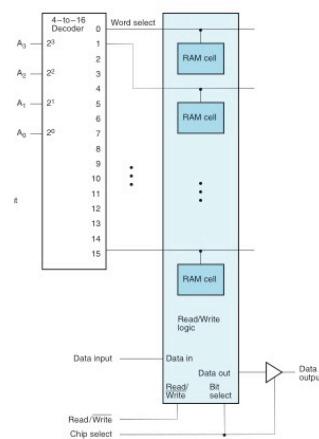
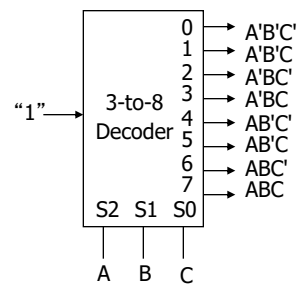
10

7.3 Decoder

11

Decoders

- Single data input, k control inputs, 2^k outputs.
- Used to select one of 2^k components:
 - RAM cells in memory
 - Registers in register file



12

2-to-4 Decoder

- Implementing a 2-to-4 decoder

Truth Table

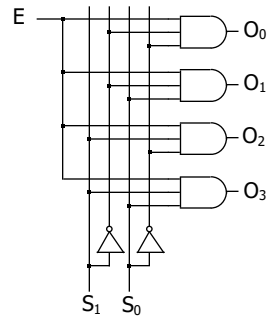
E	S ₁	S ₀	O ₃	O ₂	O ₁	O ₀
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$$O_0 = E \cdot S_1' \cdot S_0'$$

$$O_1 = E \cdot S_1' \cdot S_0$$

$$O_2 = E \cdot S_1 \cdot S_0'$$

$$O_3 = E \cdot S_1 \cdot S_0$$



We can also implement active low enable

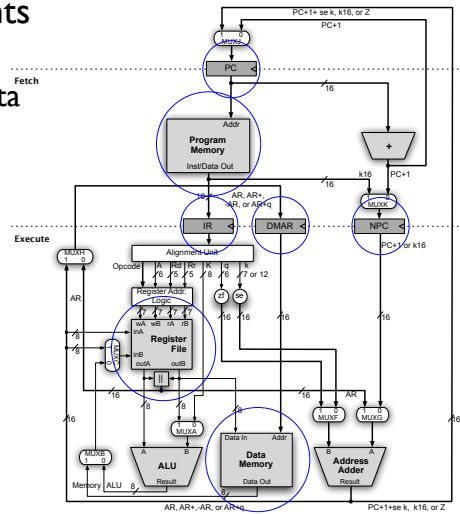
13

7.4 Memory Elements

14

Memory Elements

- Bi-stable elements used in
 - Program and Data memories
 - Registers
 - Register File

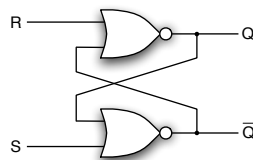


Chapter 7: Digital Components

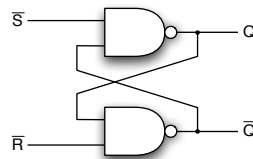
15

15

Memory Element: S-R Latch



S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	not allowed



\bar{S}	\bar{R}	Q
0	0	not allowed
0	1	1
1	0	0
1	1	hold

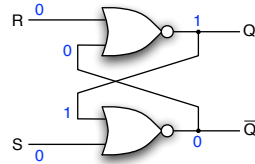
Chapter 7: Digital Components

16

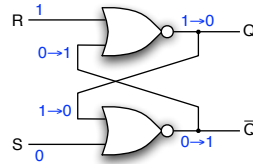
16

S-R Latch Operation

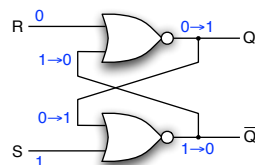
S=0, R=0 (hold), initially Q=1



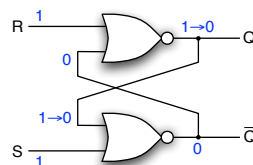
S=0, R=1 (reset), initially Q=1



S=1, R=0 (set), initially Q=0



S=1, R=1 (not allowed), initially Q=1



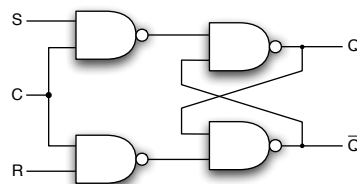
Chapter 7: Digital Components

17

17

S-R Latch with Enable

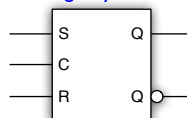
Circuit



Function Table

C	S	R	Q
0	X	X	No Change
1	0	0	No Change
1	0	1	0
1	1	0	1
1	1	1	Not allowed

Logic Symbol



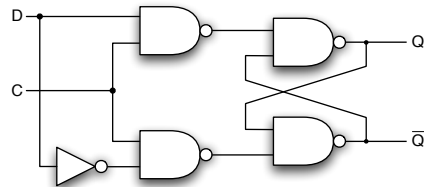
Chapter 7: Digital Components

18

18

D Latch

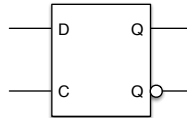
Circuit



Function Table

D	C	Q
0	1	0
1	1	1
X	0	No Change

Logic Symbol



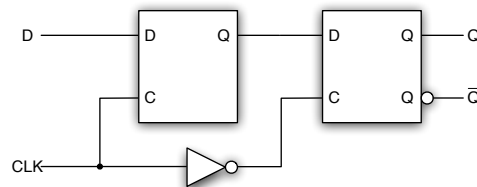
Chapter 7: Digital Components

19

19

Negative Edge-Triggered D-FF

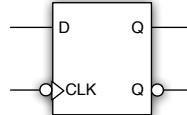
Circuit



Function Table

CLK	D	Q
↓	0	0
↓	1	1
0	0	No Change
1	1	No Change

Logic Symbol

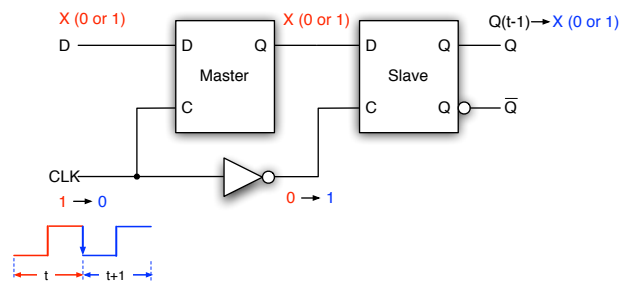


Chapter 7: Digital Components

20

20

Negative Edge-Triggered D-FF Operation

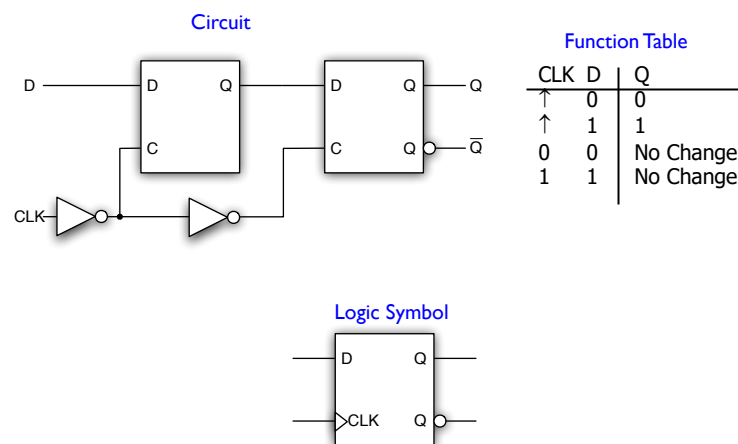


Chapter 7: Digital Components

21

21

Positive Edge-Triggered D-FF

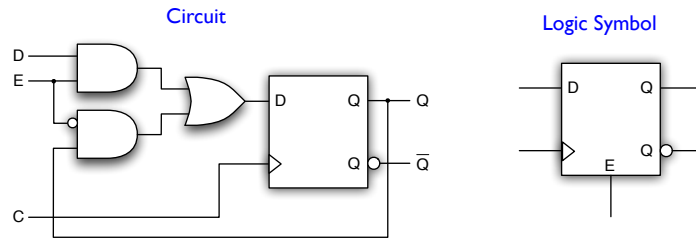


Chapter 7: Digital Components

22

22

Edge-Triggered D-FF /w Enable



Chapter 7: Digital Components

23

23

7.4 Registers

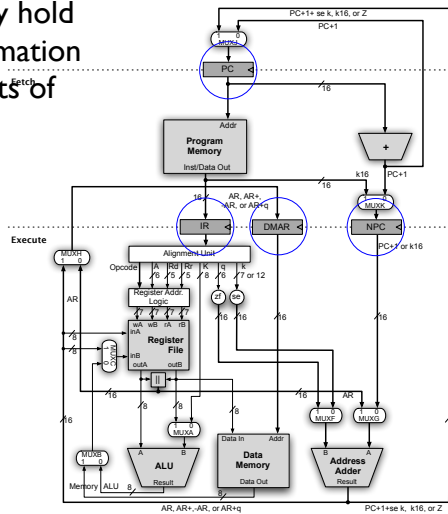
Chapter 7: Digital Components

24

24

Registers

- Used to temporary hold and separate information among various parts of the datapath.



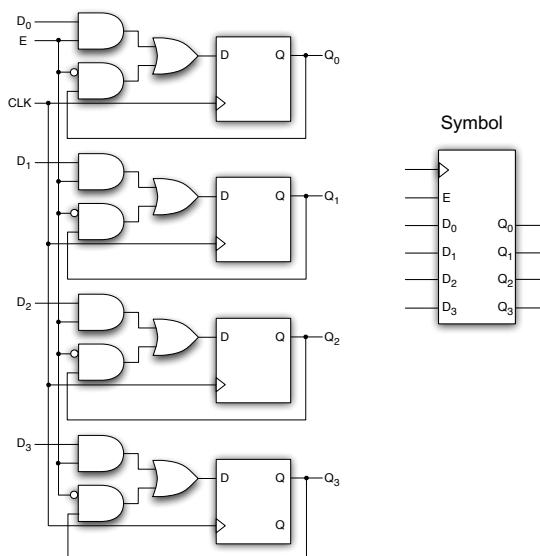
Chapter 7: Digital Components

25

25

n -bit Register /w Load Enable

A set of commonly clocked D flip-flops

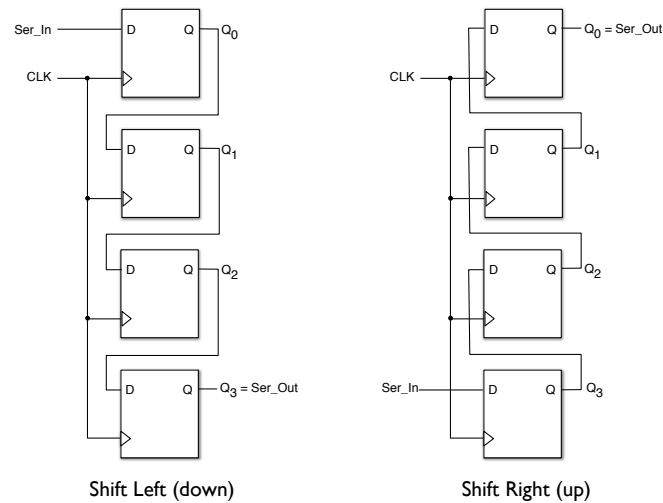


Chapter 7: Digital Components

26

26

Shift Registers

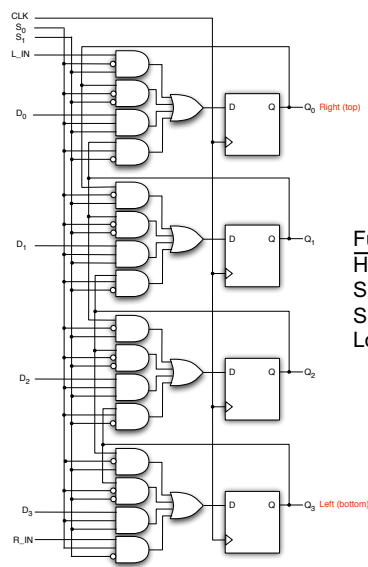


Chapter 7: Digital Components

27

27

Bi-Directional Shift Register /w Parallel Load



Function	Control Input		Next State			
	S_1	S_0	Q_3	Q_2	Q_1	Q_0
Hold	0	0	Q_3	Q_2	Q_1	Q_0
Shift right	0	1	R_IN	Q_3	Q_2	Q_1
Shift left	1	0	Q_2	Q_1	Q_0	L_IN
Load	1	1	D_3	D_2	D_1	D_0

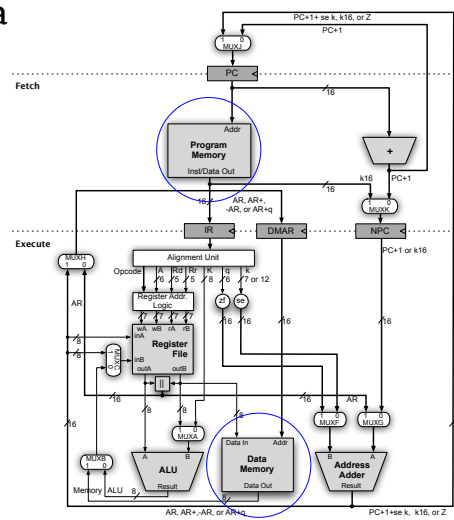
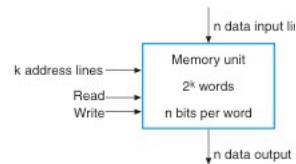
Chapter 7: Digital Components

28

28

29

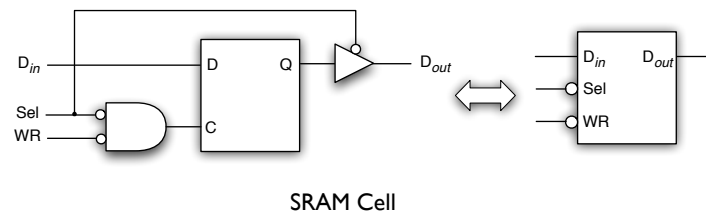
Program and Data memories



30

Memory Cell

- Static RAM (SRAM)
 - Fast
 - Practically a D-FF.
 - Used in Registers and Instruction and Data Caches.
- Dynamic RAM (DRAM)
 - Need to be refreshed periodically.
 - Used in Main Memory

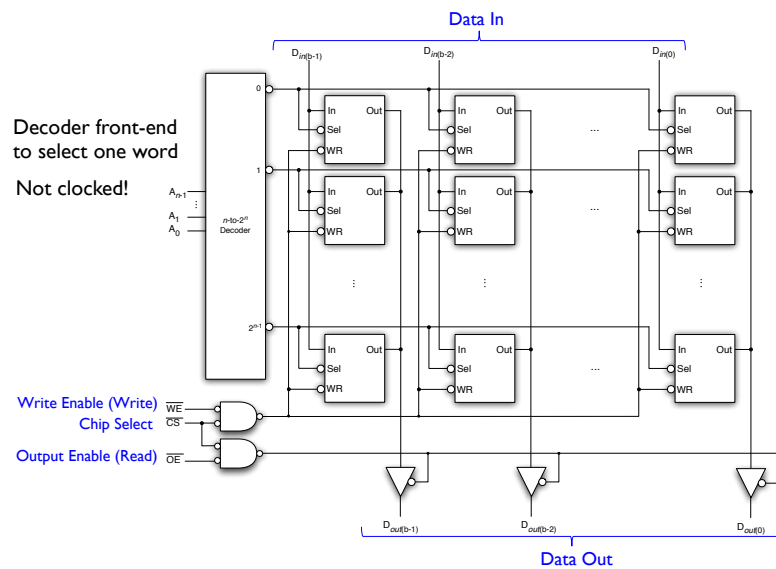


Chapter 7: Digital Components

31

31

SRAM Structure

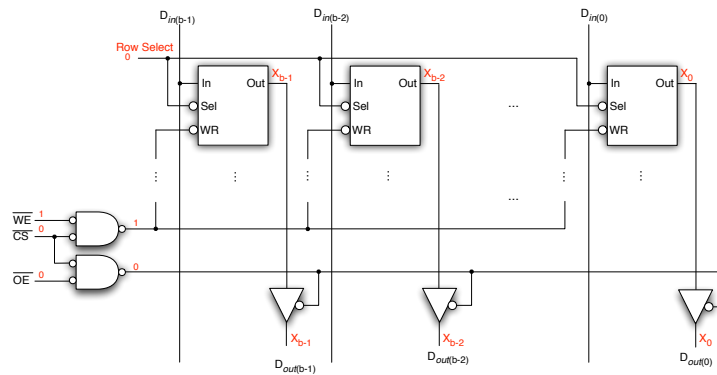


Chapter 7: Digital Components

32

32

SRAM Operation: Read

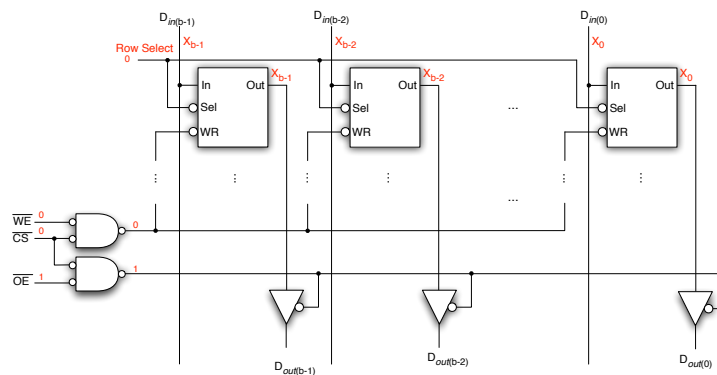


Chapter 7: Digital Components

33

33

SRAM Operation: Write



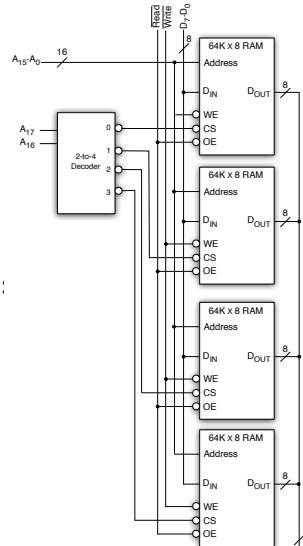
Chapter 7: Digital Components

34

34

Building Larger Memory

256K x 8 RAM
Using four 64K x 8 RAM:



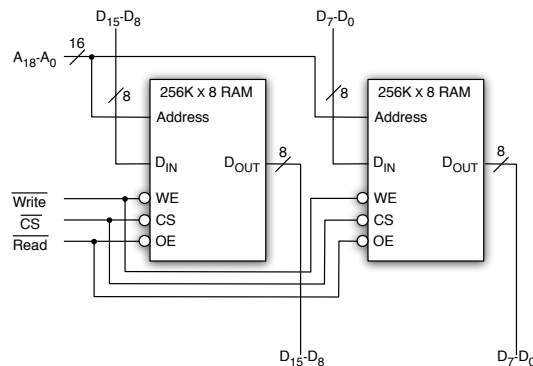
Chapter 7: Digital Components

35

35

Building Wider Memory

256K x 16 RAM
Using two 256K x 8 RAMs



Chapter 7: Digital Components

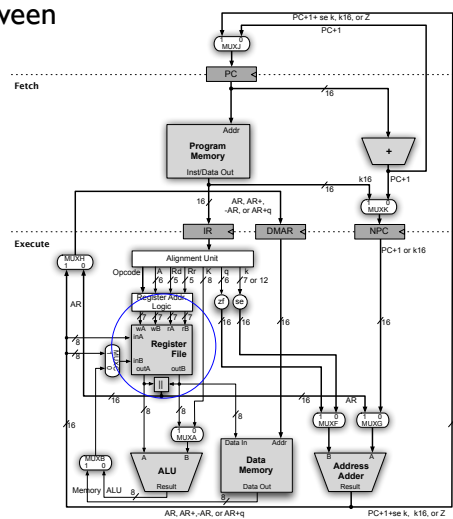
36

36

7.6 Register File

Register File

Storage buffer between
ALU and Memory



Register File

2 Read-port,
2 Write-port
X-, Y-, & Z-registers

The diagram illustrates a Register File architecture. It features a central array of registers, each with a data input (D), data output (Q), and an enable (E) signal. The registers are indexed by a 12-bit address (0 to 127). The architecture includes two read ports (rA, rB) and two write ports (wA, wB). The read ports are connected to the Q outputs of the registers. The write ports are connected to the D inputs of the registers. The enable signals (E) are connected to the outputs of two 12-to-1 multiplexers (MUX A and MUX B), which are controlled by the Register Identifiers (rA, rB, wA, wB). The Register Identifiers are also connected to two 12-to-1 decoders (Decoder A and Decoder B), which generate the RF_wA and RF_wB write signals. The data inputs (inA, inB) are connected to the D inputs of the registers. The data outputs (outA, outB) are connected to the Q outputs of the registers. The clock signal (CLK) is connected to the clock inputs of the registers.

Register Identifiers

Read

Write

Data In

CLK

inA

inB

RF_wB

RF_wA

Write signals

outB

outA

Data Out

Chapter 7: Digital Components

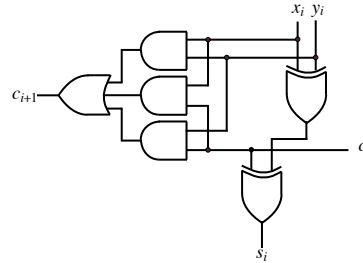
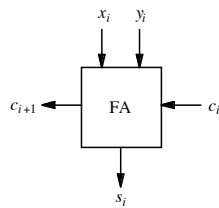
39

39

Basic ALU

40

Full Adder



x_i	y_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$x_i \backslash y_i$	00	01	11	10
0		1		1
1	1		1	

$$s_i = x_i \oplus y_i \oplus c_i$$

$x_i \backslash y_i$	00	01	11	10
0			1	
1		1	1	1

$$c_{i+1} = x_i y_i + (x_i \oplus y_i) c_i$$

or

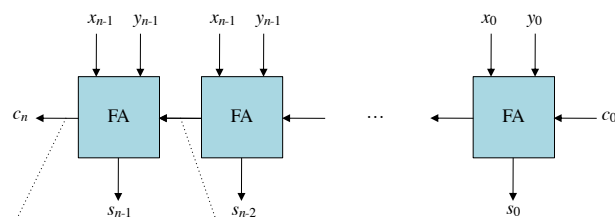
$$c_{i+1} = x_i y_i + (x_i + y_i) c_i$$

Chapter 7: Digital Components

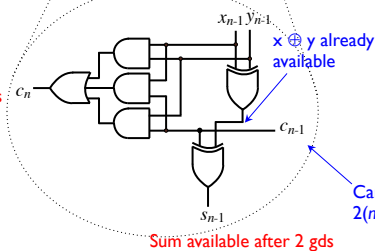
41

41

n-bit Ripple Carry Adder



Carry-out
available
after 2 gds



Assuming x and y are available at the same time.

Based on $c_{i+1} = x_i y_i + (x_i + y_i) c_i$

Valid result available after $2n$ gds!

Chapter 7: Digital Components

42

42

Adder/Subtractor

- With RCA and a complementor, we can perform a variety of operations:
 - $x+y$
 - $x+y'$
 - $x+y'+1$
 - $0+y$
 - y'
 - $y'+1$

The diagram illustrates a ripple-carry adder/subtractor circuit. It consists of three Full Adders (FA) connected in series. The inputs to each FA are x_{n-1} , y_{n-1} , and a carry-in (C_n). The outputs are S_{n-1} , S_{n-2} , and S_0 . The carry-out of one stage is the carry-in for the next stage. The carry-in C_n is determined by the operation being performed: $C_n = y'c + yc'$.

y	c	result
0	0	0
0	1	1
1	0	1
1	1	0

Result = $y'c + yc'$

Chapter 7: Digital Components 43

43

Condition Codes in SREG

Sign-bit (S)

N	V	S
0	0	0
0	1	1
1	0	1
1	1	0

Correct
Incorrect!
Correct
Incorrect!

Rule for Overflow (V)
Carry-in, carry-out
No carry-in, no carry out
Otherwise, overflow!

Carry-out	Bit <i>n</i> -1	Carry-in
0	0	1
0	1	1
1	0	0
1	1	0

Chapter 7: Digital Components

44

44

Questions?

