Esercizi 01

Computer Design Basics

Chapter 8 - Solutions

1) Given a 4-bits barrel shifter, find the output Y for each of the following inputs applied to (S_1, S_0, D_3, D_2, D_1, D_0):

(a) 110101 **(b) 101011**
(c) 011010 **(d) 001101**

Solution:

The bits S1 e S0 indicate the number of positions to shift. Therefore, we have:

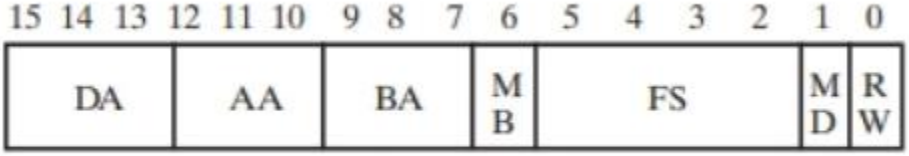
	S_1	S_0	Positions	$D_3D_2D_1D_0$	Y
(a)	1	1	3	0101	1010
(b)	1	0	2	1011	1110
(c)	0	1	1	1010	0101
(d)	0	0	0	1101	1101

2) Specify the 16-bits control word to be applied to the datapath to implement the following microoperations:

(a) R3 ←Data in **(b) R4 ← 0**
(c) R1 ← sr R4 **(d) R3 ← R3 + 1**
(e) R2 ←sl R2 **(f) R1 ← R2 ⊕ R4**
(g) R7 ← R1 + R3 **(h) R4 ← R5 – Constant in**

Solution:

The control word is composed of the following fields:



(b) Control word

By using the Table 8-5 showing the encoding of control word:

TABLE 8-5
Encoding of Control Word for the Datapath

DA, AA, BA		MB		FS		MD		RW	
Function	Code	Function	Code	Function	Code	Function	Code	Function	Code
$R0$	000	Register	0	$F = A$	0000	Function	0	No Write	0
$R1$	001	Constant	1	$F = A + 1$	0001	Data in	1	Write	1
$R2$	010			$F = A + B$	0010				
$R3$	011			$F = A + B + 1$	0011				
$R4$	100			$F = A + \overline{B}$	0100				
$R5$	101			$F = A + \overline{B} + 1$	0101				
$R6$	110			$F = A - 1$	0110				
$R7$	111			$F = A$	0111				
				$F = A \wedge B$	1000				
				$F = A \vee B$	1001				
				$F = A \oplus B$	1010				
				$F = \overline{A}$	1011				
				$F = B$	1100				
				$F = \text{sr } B$	1101				
				$F = \text{sl } B$	1110				

The required control words are:

	DA	AA	BA	MB	FS	MD	RW
(a) R3←Data in	011	---	---	-	----	1	1
(b) R4←0	100	000	000	0	1010	0	1
	The registers A e B are used, the XOR is applied (→0 with the same inputs)						
(c) R1←sr R4	001	---	100	0	1101	0	1
(d) R3 ← R3 + 1	011	011	---	-	0001	0	1
(e) R2 ← sl R2	010	---	010	0	1110	0	1
(f) R1 ← R2 ⊕ R4	001	010	100	0	1010	0	1
(g) R7 ← R1 + R3	111	001	011	0	0010	0	1
(h) R4 ← R5 – Constant in	100	101	---	1	0101	0	1

3) Given the following 16-bits control words for the datapath, determine a) the microoperation that is executed and b) the contents of the registers for each control word. Assume that the registers contain the value of their index (e.g., the register R5 contains 05 in hexadecimal). Assume that Constant in has value has 6 and Data In has value 1B.

(a) 101 100 101 0 1000 0 1
(b) 110 010 100 0 0101 0 1
(c) 101 110 000 0 1100 0 1
(d) 101 000 000 0 0000 0 1
(e) 100 100 000 1 1101 0 1
(f) 011 000 000 0 0000 1 1

Solution:

Keep in mind the format of the control word and use the Table 8-5 showing the encodings

	DA	AA	BA	MB	FS	MD	RW	
(a)	101	100	101	0	1000	0	1	R5 ←R4 ∧ R5
(b)	110	010	100	0	0101	0	1	R6 ← R2 – R4
(c)	101	~~110~~	000	0	1100	0	1	R5 ← R0
(d)	101	000	~~000~~	0	0000	0	1	R5 ← R0
(e)	100	~~100~~	~~000~~	1	1101	0	1	R4 ←sr Constant
(f)	011	~~000~~	~~000~~	000	0000	1	1	R3 ←Data in

(a)	R5 ←R4 ∧ R5	AND: R5 = 0000 0100
(b)	R6 ← R2 – R4	R6 = 1111 1110
(c)	R5 ← R0	R5 = 0000 0000
(d)	R5 ← R0	R5 = 0000 0000
(e)	R4 ←sr Constant	R4 = 0001 1000
(f)	R3 ←Data in	R3 = 0001 1011

4) A computer has a 32-bits instruction format consisting of the following: opcode (6 bits), 2 registers addresses (5 bits per each), and one immediate operand/registers addresses (16 bits).

a) What is the maximum number of operation possible?

b) How many registers can be addressed?

c) What is the maximum value of an immediate operand (unsigned) that can be provided?

d) What is the range of an immediate operand (unsigned) that can be provided if we assume that the operands are signed 2s complement binary numbers and the 15^{th} bit is the one associated with the sign?

Solution:

a) opcode → 6 bits; so the number of possible operations are → $2^6 = 64$

b) 1 x register → 5 bits; number of registers → $2^5 = 32$

c) The range of an immediate operand [0; 2^{32} -1]. The maximum value corresponds to 65535 = $(1111\ 1111\ 1111\ 1111)_2$ → FFFF

d) from $(2^{15} – 1)$ to -2^{15} → +32767 a -32768

5) A computer has a memory unit with 32 bits instruction and a register file with 64 registers. The instruction set consists of 130 different operations. The instruction format is the following: opcode, a register file address, and the immediate operand part. Each instruction is stored in one word of memory.

a) How many bits are necessary for the opcode part of the instruction?

b) How many bits are left for the immediate operand?

c) If the immediate operand is used as an unsigned address to memory, what is the maximum number of words that can be addressed in memory?

d) What are the largest and the smallest algebraic values of signed 2s complement binary numbers that can be accommodated as an immediate operand?

Solution:

a) 130 operations → $\log_2 130 = 7.02$ → 8 bits

b) 64 registers → bits to address registers = 6 bits; bits for immediate operands = 18 bits

c) $2^{18} = 262144$ in the range $[0; 2^{18}-1]$

d) from $(2^{17} – 1)$ to -2^{17} → +131071 e -131072

6) A single-cycle computer executes the five instructions in the following tables (a) and (b).

a) Complete the table

Solution

Instruction Register Tranfer	DA	AA	BA	MB	FS	MD	RW	MW	PL	JB
R[0] ← R[7] ⊕ R[3]	000	111	011	0	1010	0	1	0	0	x
R[1] ← M[R[4]]	001	100	xxx	X	xxxx	1	1	0	0	x
R[2] ← R[5] + 2	010	101	xxx	1	0010	0	1	0	0	x
R[3] ← sl R[6]	011	xxx	110	0	1110	0	1	0	0	x

**if R[4] = 0 then
PC ← PC + se AD
else
PC ← PC + 1**
	xxx	100	xxx	x	0000	x	0	0	1	0

b) Complete the table.

Instruction Register Transfer	Operation Code	DR	SA	SB or Operand
R[0] ←R[7] + R[6]	000 0010	000	111	110
R[1] ← R[5] - 1	000 0110	001	110	000
R[2] ← sl R[4]	000 1110	010	000	100
R[3] ← $\overline{R[3]}$	000 1011	011	011	000
R[4] ← R[2] ∨ R[1]	000 1001	100	010	001

7) Simulate the following sequence of instructions, assuming that each register initially contains contents equal to its index (e.g., R1 contains 1). Report the binary value of each instruction and the content of any register changed by the instruction.

Solution

Instruction	**Code**	**Registers/Memory changed**
ADD R0, R1, R2	000 0010 000 001 010	
SUB R3, R4, R5	000 0101 011 100 101	R0 = 3
SUB R6, R7, R0	000 0101 110 111 000	R3 = -1
ADD R0, R0, R3	000 0010 000 000 011	R6 = 4
SUB R0, R0, R6	000 0101 000 000 110	R0 = 2
ST R7, R0	010 0000 000 111 000	R0 = -2
LD R7, R6	001 0000 111 110 000	M[7] = -2
ADI R0, R6, 2	100 0010 000 110 010	R7 = M[4]
ADI R3, R6, 3	100 0010 011 110 011	R0 = 6
		R3 = 7

8) How many Mbyte is possible to address with 16 bit?, 20 bit?, 24 bit?, 32 bit?, 64 bit?

Solution:

Given that a 1KByte memory corresponds to 2^{10} = 1024 bytes, we can achieve that 1Mbyte = 1Kbyte X 1Kbyte = 2^{10} X 2^{10} = 2^{20} bytes.

We can compute:

16 bit -> 2^{16} bytes = $2^{16}/2^{20}$ Mbyte = 1/16 MB = 64 Kbyte

20 bit -> 2^{20} bytes = $2^{20}/2^{20}$ Mbyte = 1 Mbyte

24 bit -> 2^{24} bytes =$2^{24}/2^{20}$ Mbyte = 2^{4} Mbyte = 16 Mbyte

32 bit -> 2^{32} bytes = $2^{32}/2^{20}$ Mbyte = 2^{12} Mbyte = 4096 Mbyte = 4 Gbyte

64 bit -> 2^{64} bytes = $2^{64}/2^{20}$ Mbyte = 2^{44} Mbyte = 2^{34} Gbyte = 2^{24} Tbyte