

# Esercizi 01

## Computer Design Basics

### Chapter 8

1) Determinare l'output Y di un 4-bit barrel shifter per ognuno dei seguenti valori di ingresso ( $S_1, S_0, D_3, D_2, D_1, D_0$ ):

- |            |            |
|------------|------------|
| (a) 110101 | (b) 101011 |
| (c) 011010 | (d) 001101 |

**Soluzione:**

I bit  $S_1$  e  $S_0$  determinano il numero di posizioni da shiftare. Quindi:

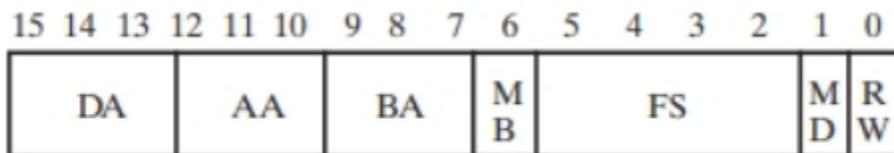
	$S_1$	$S_0$	Posizioni	$D_3D_2D_1D_0$	Y
(a)	1	1	3	0101	1010
(b)	1	0	2	1011	1110
(c)	0	1	1	1010	0101
(d)	0	0	0	1101	1101

2) Specificare la control word da 16-bit che deve essere applicata per implementare le seguenti microoperazioni:

- |                                    |   |
|------------------------------------|---|
| (a) $R3 \leftarrow \text{Data in}$ | (b) $R4 \leftarrow 0$                       |
| (c) $R1 \leftarrow \text{sr } R4$  | (d) $R3 \leftarrow R3 + 1$                  |
| (e) $R2 \leftarrow \text{sl } R2$  | (f) $R1 \leftarrow R2 \oplus R4$            |
| (g) $R7 \leftarrow R1 + R3$        | (h) $R4 \leftarrow R5 - \text{Constant in}$ |

**Soluzione:**

Ricordiamo il formato del control word:



(b) Control word

E la tabella delle codifiche corrispondenti:

□ 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	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	Write	1
R2	010		$F = A + B$	0010			
R3	011		$F = A + B + 1$	0011			
R4	100		$F = A + \bar{B}$	0100			
R5	101		$F = A + \bar{B} + 1$	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			
			$F = \bar{A}$	1011			
			$F = B$	1100			
			$F = sr B$	1101			
			$F = sl B$	1110			

Da cui le control word richieste sono:

	<u>DA</u>	<u>AA</u>	<u>BA</u>	<u>MB</u>	<u>FS</u>	<u>MD</u>	<u>RW</u>
(a) $R3 \leftarrow \text{Data in}$	011	---	---	-	----	1	1
(b) $R4 \leftarrow 0$	100	000	000	0	1010	0	1
Si usa lo stesso registro per A e B, e si applica uno XOR ( $\rightarrow 0$ con input uguali)							
(c) $R1 \leftarrow sr R4$	001	---	100	-	1101	0	1
(d) $R3 \leftarrow R3 + 1$	011	011	---	-	0001	0	1
(e) $R2 \leftarrow sl R2$	010	---	010	-	1110	0	1
(f) $R1 \leftarrow R2 \oplus R4$	001	010	100	0	1010	0	1
(g) $R7 \leftarrow R1 + R3$	111	001	011	0	0010	0	1
(h) $R4 \leftarrow R5 - \text{Constant in}$	100	101	---	1	0101	0	1

3) Date le seguenti control word a 16 bit, determinare a) le microoperazioni eseguite e b) il contenuto dei registri dopo ogni control word. Si assume che i registri contengano il valore del loro indice (ad esempio R5 contiene 05 in esadecimale). Si assume che la Costante abbia valore 6 e Data In abbia valore 1B.

- (a) 101 100 101 0 1000 0 1  
 (c) 101 110 000 0 1100 0 1  
 (e) 100 100 000 1 1101 0 1

- (b) 110 010 100 0 0101 0 1  
 (d) 101 000 000 0 0000 0 1  
 (f) 011 000 000 0 0000 1 1

**Soluzione:**

Si ricorda il formato del control word e la tabella di codifica

	<u>DA</u>	<u>AA</u>	<u>BA</u>	<u>MB</u>	<u>FS</u>	<u>MD</u>	<u>RW</u>	
(a)	101	100	101	0	1000	0	1	$R5 \leftarrow R4 \wedge R5$
(b)	110	010	100	0	0101	0	1	$R6 \leftarrow R2 - R4$
(c)	101	<del>110</del>	000	0	1100	0	1	$R5 \leftarrow R0$
(d)	101	000	<del>000</del>	0	0000	0	1	$R5 \leftarrow R0$
(e)	100	<del>100</del>	<del>000</del>	1	1101	0	1	$R4 \leftarrow sr \text{ Constant}$
(f)	011	000	000	000	0000	1	1	$R3 \leftarrow \text{Data in}$

(a)	$R5 \leftarrow R4 \wedge R5$	AND: $R5 = 0000 \ 0100$
(b)	$R6 \leftarrow R2 - R4$	$R6 = 1111 \ 1110$
(c)	$R5 \leftarrow R0$	$R5 = 0000 \ 0000$
(d)	$R5 \leftarrow R0$	$R5 = 0000 \ 0000$
(e)	$R4 \leftarrow sr \text{ Constant}$	$R4 = 0001 \ 1000$
(f)	$R3 \leftarrow \text{Data in}$	$R3 = 0001 \ 1011$

4) Un computer ha un'istruzione a 32-bit divisa in: opcode (6 bits), due indirizzi di registri (5 bits ognuno), e un operando immediato/indirizzo di registro (16 bits).

a) Qual è il numero massimo di operazioni che possono essere specificate?

b) Quanti registri possono essere indirizzati?

c) Qual è il range di un operando immediato (unsigned) che può essere fornito?

d) Qual è il range di un operando immediato (signed) che può essere fornito assumendo che gli operandi hanno una rappresentazione a complemento a due e che il bit 15 è quello di segno?

**Soluzione:**

a) opcode  $\rightarrow$  6 bits; operazioni possibili  $\rightarrow 2^6 = 64$

b) 1 x registro  $\rightarrow$  5 bits; registri possibili  $\rightarrow 2^5 = 32$

c) Massimo valore operando: FFFF  $\rightarrow$  65536 simboli, ovvero da 0 a 65535

d) da  $(2^{15} - 1)$  a  $-2^{15} \rightarrow +32767$  a  $-32768$

5) Un computer ha una unità di memoria con istruzioni a 32-bit e un file di registro con 64 registri. Il set di istruzioni consiste in 130 operazioni. Il formato di istruzioni è il seguente: opcode, indirizzo di registro e operando immediato. Ogni istruzione è immagazzinata in 1 word.

a) Quanti bits sono necessari per l'opcode?

b) Quanti bit rimangono per l'operando immediato?

c) Se l'operando immediato è usato come un indirizzo di memoria unsigned, qual è il massimo numero di parole che possono essere indirizzate in memoria?

d) Qual è il più grande e il più piccolo valore a complemento a due che un operando immediato può assumere?

**Soluzione:**

a) 130 operazioni  $\rightarrow \log_2 130 = 7.02 \rightarrow 8$  bits

b) 64 registri  $\rightarrow$  bits per indirizzare registri = 6 bits; bits per l'operando immediato = 18 bits

c)  $2^{18} = 262144$

d) +131071 e -131072

**6) Un computer a ciclo singolo esegue 5 operazioni come in tabella.**

TABLE 8-8

Instruction Specifications for the Simple Computer

Instruction	Opcode	Mnemonic	Format	Description	Status Bits
Move A	0000000	MOVA	RD, RA	$R[DR] \leftarrow R[SA]^*$	N, Z
Increment	0000001	INC	RD, RA	$R[DR] \leftarrow R[SA] + 1^*$	N, Z
Add	0000010	ADD	RD, RA, RB	$R[DR] \leftarrow R[SA] + R[SB]^*$	N, Z
Subtract	0000101	SUB	RD, RA, RB	$R[DR] \leftarrow R[SA] - R[SB]^*$	N, Z
Decrement	0000110	DEC	RD, RA	$R[DR] \leftarrow R[SA] - 1^*$	N, Z
AND	0001000	AND	RD, RA, RB	$R[DR] \leftarrow R[SA] \wedge R[SB]^*$	N, Z
OR	0001001	OR	RD, RA, RB	$R[DR] \leftarrow R[SA] \vee R[SB]^*$	N, Z
Exclusive OR	0001010	XOR	RD, RA, RB	$R[DR] \leftarrow R[SA] \oplus R[SB]^*$	N, Z
NOT	0001011	NOT	RD, RA	$R[DR] \leftarrow \overline{R[SA]}^*$	N, Z
Move B	0001100	MOVB	RD, RB	$R[DR] \leftarrow R[SB]^*$	
Shift Right	0001101	SHR	RD, RB	$R[DR] \leftarrow sr R[SB]^*$	
Shift Left	0001110	SHL	RD, RB	$R[DR] \leftarrow sl R[SB]^*$	
Load Immediate	1001100	LDI	RD, OP	$R[DR] \leftarrow zf OP^*$	
Add Immediate	1000010	ADI	RD, RA, OP	$R[DR] \leftarrow R[SA] + zf OP^*$	N, Z
Load	0010000	LD	RD, RA	$R[DR] \leftarrow M[SA]^*$	
Store	0100000	ST	RA, RB	$M[SA] \leftarrow R[SB]^*$	
Branch on Zero	1100000	BRZ	RA, AD	if $(R[SA] = 0)$ $PC \leftarrow PC + se AD$ , N, Z if $(R[SA] \neq 0)$ $PC \leftarrow PC + 1$	
Branch on Negative	1100001	BRN	RA, AD	if $(R[SA] < 0)$ $PC \leftarrow PC + se AD$ , N, Z if $(R[SA] \geq 0)$ $PC \leftarrow PC + 1$	
Jump	1110000	JMP	RA	$PC \leftarrow R[SA]$	

\* For all of these instructions,  $PC \leftarrow PC + 1$  is also executed to prepare for the next cycle.

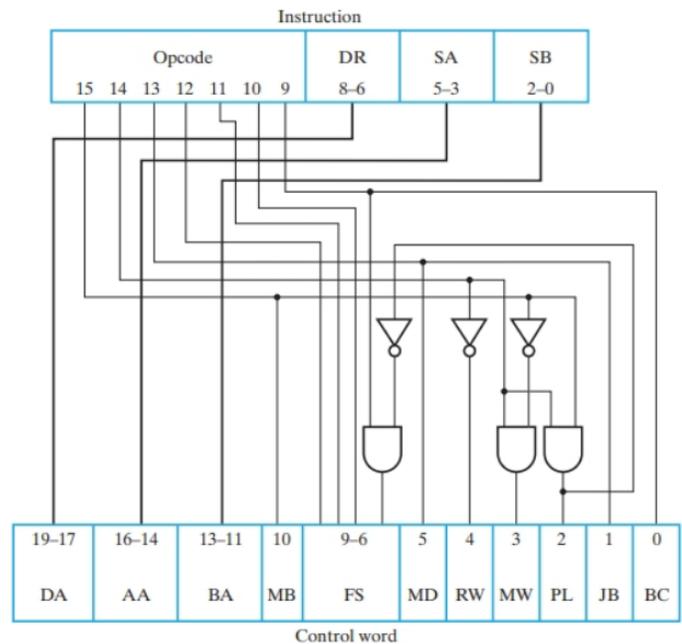


FIGURE 8-16 Diagram of Instruction Decoder

**NOTA:** i bit 9, 10, 11, 12 di OPCODE vengono usati per FS. Il bit 9 è però in AND con NOT del bit PL (che è 1 se bit 15 e 15 sono entrambi 1)

**TABLE 8-10**  
**Truth Table for Instruction Decoder Logic**

Instruction Function Type	Instruction Bits				Control-Word Bits						
	15	14	13	9	MB	MD	RW	MW	PL	JB	BC
Function-unit operations using registers	0	0	0	X	0	0	1	0	0	X	X
Memory read	0	0	1	X	0	1	1	0	0	X	X
Memory write	0	1	0	X	0	X	0	1	0	X	X
Function-unit operations using register and constant	1	0	0	X	1	0	1	0	0	X	X
Conditional branch on zero ( <i>Z</i> )	1	1	0	0	X	X	0	0	1	0	0
Conditional branch on negative ( <i>N</i> )	1	1	0	1	X	X	0	0	1	0	1
Unconditional jump	1	1	1	X	X	X	0	0	1	1	X

a) Completare la tabella

Instruction Register Transfer	DA	AA	BA	MB	FS	MD	RW	MW	PL	JB
$R[0] \leftarrow R[7] \oplus R[3]$	000	111	011	0	1010	0	1	0	0	x
$R[1] \leftarrow M[R[4]]$	001	100	xxx	X	xxxx	1	1	0	0	x
$R[2] \leftarrow R[5] + 2$	010	101	xxx	1	0010	0	1	0	0	x
$R[3] \leftarrow sl R[6]$	011	xxx	110	0	1110	0	1	0	0	x
if $R[4] = 0$ then $PC \leftarrow PC + se AD$ else $PC \leftarrow PC + 1$	xxx	100	xxx	x	0000	x	0	0	1	0

b) Completare la tabella.

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

7) Simulare la sequenza di istruzioni seguente assumendo che ogni registro contenga il valore del suo indice. Riportare il valore binario dell'istruzione il contenuto del registro in questione.

Instruction	Code	Registers/Memory changed
ADD R0, R1, R2	000 0011 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 0011 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 000 110 000	R7 = M[4]
ADI R3, R6, 3	100 0010 011 110 011	R0 = 6 R3 = 7

8) Quanti Mbyte si riescono ad indirizzare con 16 bit?, 20 bit?, 24 bit?, 32 bit?, 64 bit?

**Soluzione:**

In generale avendo a disposizione N bit di indirizzo la memoria indirizzabile risulta essere  $2^N$  bytes. Ora, sapendo che 1KByte di memoria corrisponde a  $2^{10} = 1024$  bytes, si ha che 1Mbyte = 1Kbyte X 1Kbyte =  $2^{10} \times 2^{10} = 2^{20}$  bytes.

Quindi per i casi proposti si hanno le seguenti soluzioni:

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