

CPU: registro Instruction Pointer (**IP**); registro generale (**AL**, 8 bit); registro indice (**SI**, 16 bit); registri interni (**IR,T**); ALU

RAM: capacità di 3072 celle di memoria da 1 byte ciascuna, dall'indirizzo 0400_{hex} a 0FFF_{hex}

ROM: capacità di 768 celle di memoria da 1 byte, da 0100_{hex} a 03FF_{hex}

I/O: 128 indirizzi di input, da 00_{hex} a 7F_{hex}, e 128 indirizzi di output, da 80_{hex} a FF_{hex}

Esempio:

0100	BE 00 01	MOV SI, 0100
0103	B0 00	MOV AL, 00
0105	88 84 00 05	MOV [SI+0500], AL
0109	4E	DEC SI
010A	75 F9	JNZ 0105
010C	CD 20	INT 20

Mediante addizione in complemento a 2

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 113

Architettura semplificata

Read memory: la ROM, attivata dall'indirizzo presente in AB, seleziona la sua cella di indirizzo 0109 e, dato il valore della linea R/W (=1), pone il contenuto della cella in DB

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 115

Architettura semplificata

Fetch (1): la CPU emette in AB il contenuto di IP e pone la linea R/W a 1 per chiedere la lettura della prossima istruzione da eseguire

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 114

Architettura semplificata

Fetch (2): la CPU legge il valore presente in DB, lo pone in IR e si prepara per la prossima parola di istruzione incrementando IP

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 116

Architettura semplificata

Decodifica: il codice operativo dell'istruzione corrente (**4E**) informa la CPU che essa deve decrementare di 1 il contenuto di SI.

L'istruzione è lunga un solo byte, che ne rappresenta il **codice operativo**.

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 117

Architettura semplificata

L'istruzione DEC SI è stata completata mediante :

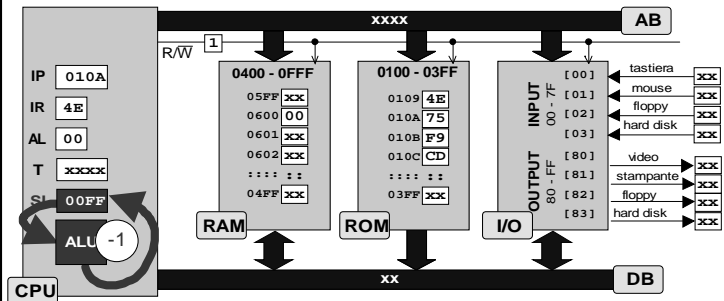
- **Fetch** (fase 1): emetti IP in AB
 - *Richiesta di lettura in memoria*
- **Fetch** (fase 2): scrivi in IR il valore fornito da DB, incrementa IP, decodifica l'istruzione in IR
- **Execute** (fase 1): decrementa SI
 - *In questo caso non sono necessarie altre sottofasi*

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 119

Architettura semplificata



Execute (1): la CPU invia all'ALU il contenuto di SI (0100) ed impone l'operazione di decremento di un'unità.

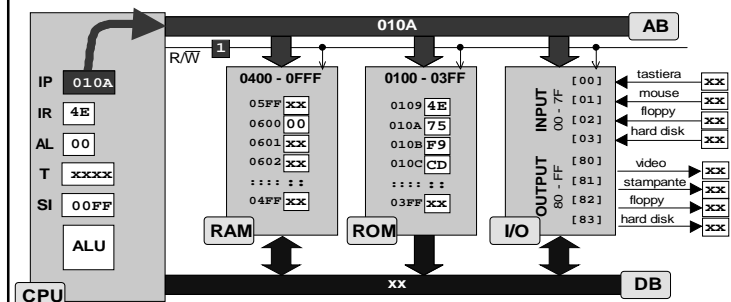
Il risultato (00FF) viene salvato nello stesso registro SI.

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 118

Architettura semplificata



Fetch (1): la CPU emette in AB il contenuto di IP e pone la linea R/W a 1 per leggere la prossima istruzione da eseguire

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 120

Architettura semplificata

Read memory: la ROM, attivata dall'indirizzo presente in AB, seleziona la sua cella di indirizzo 010A e, dato il valore della linea R/W (=1), pone il contenuto della cella in DB

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 121

Architettura semplificata

Decodifica: il codice operativo dell'istruzione corrente (**75**) informa la CPU che l'istruzione è condizionata dal valore del bit di zero (**flag Z**).

Z=1 → prosegui normalmente (*jump if not zero*)

Z=0 → somma l'operando (*displacement*) ad IP

L'operazione si può effettuare su 16 bit in complemento a 2 (oppure sugli 8 bit della parte bassa dei dati trascurando l'eventuale riporto)

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 123

Architettura semplificata

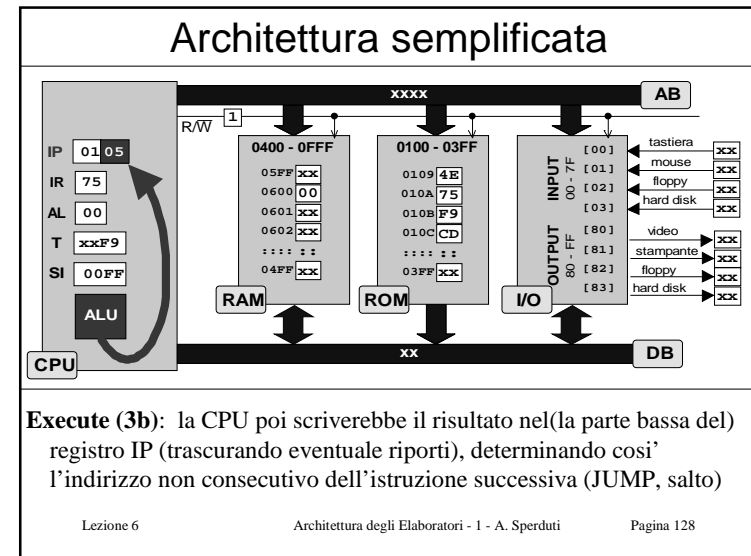
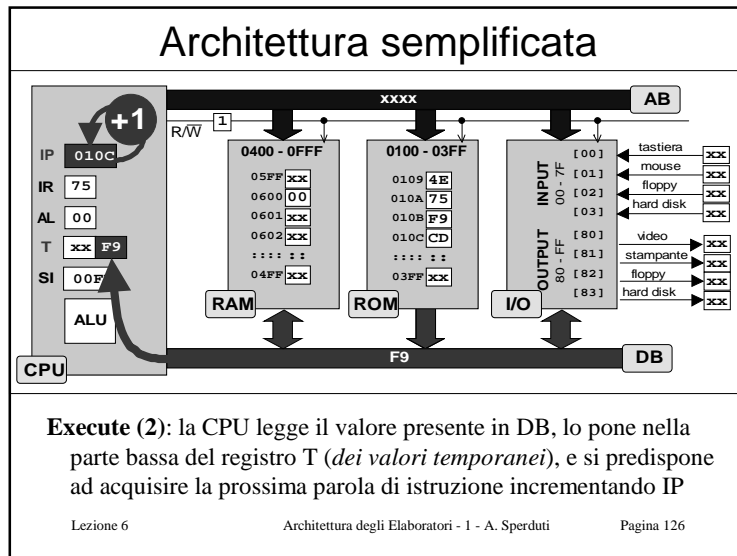
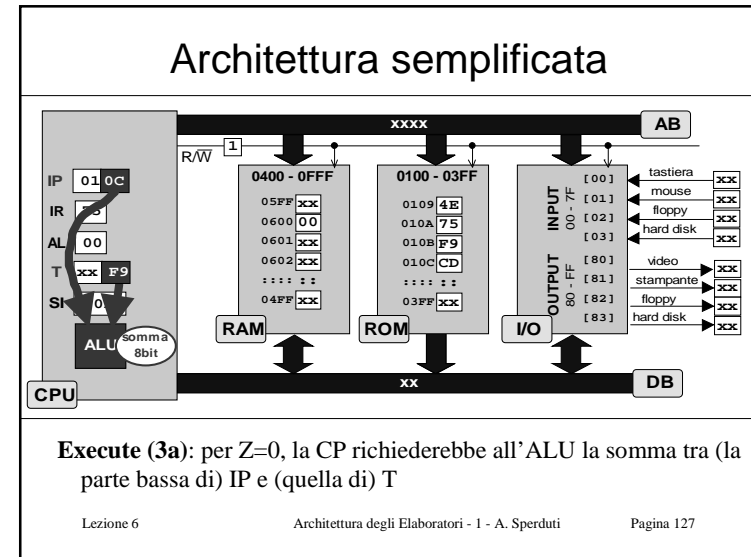
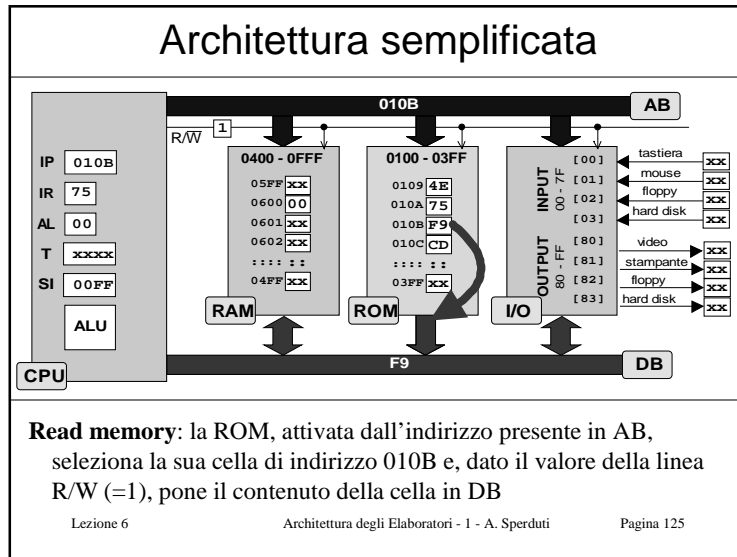
Fetch (2): la CPU legge il valore presente in DB, lo pone in IR e si prepara per la prossima parola di istruzione incrementando IP

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 122

Architettura semplificata

Execute (1): l'istruzione ha 1 operando, dunque la CPU emette il contenuto di IP in AB e pone la linea R/W a 1, per leggere l'operando (displacement)

Lezione 6 Architettura degli Elaboratori - 1 - A. Sperduti Pagina 124



Architettura semplificata

L'istruzione JNZ F9 è stata completata mediante:

- **Fetch** (1): emetti IP in AB
 - *Richiesta di lettura in memoria*
- **Fetch**(2): scrivi in IR il valore in DB, incrementa IP, effettua decodifica
- **Execute** (1): emetti IP in AB
 - *Richiesta di lettura in memoria*
- **Execute** (2): scrivi in T il valore in DB, incrementa IP
- **Execute** (3): verifica Z, se Z=0 allora IP + T → IP

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 129

Altro Esempio

Sequenza fetch/execute:

- IP = 0100 Pone il contenuto di IP nell'address bus e la linea R/W ad 1. Carica il codice operativo (opcode) 9A_{hex} nel registro dati dal data bus. Incrementa IP (IP=0101). Muove il dato dal registro dati al registro IR. Decodifica il codice operativo (la decodifica segnala che è necessario un operando).
- IP = 0101 Pone il contenuto di IP nell'address bus e la linea R/W ad 1. Carica il byte = 23_{hex} nel registro dati. Incrementa IP (IP=0102); punta alla istruzione ADC \$2043). Muove il dato nell'accumulatore ACC.
- IP = 0102 Pone il contenuto di IP nell'address bus e la linea R/W ad 1. Carica il codice operativo 8A_{hex} nel registro dati dal data bus. Incrementa IP (IP=0103). Muove il dato dal registro dati al registro IR. Decodifica il codice operativo (la decodifica segnala che i prossimi 2 byte in memoria contengono l'indirizzo del dato da utilizzare).

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 131

Altro Esempio

Architettura con accumulatore (Assembler 6502):

- **il programma** (notazione esadecimale: (hex)adecimal)

INDIRIZZI	CONTENUTO	PROGRAMMA	COMMENTO
0100	9A 23	LDA #23	carica il byte con valore 23 _{hex} nell'accumulatore
0102	8A 43 20	ADC \$2043	aggiungi il byte contenuto all'indirizzo 2043 _{hex} , e se la flag di riporto è on, aggiungi 1
0105	84 44 20	STA \$2044	memorizza il risultato all'indirizzo 2044h

- **la memoria (byte per byte):** memorizzazione dati Little Endian

INDIRIZZI	CONTENUTO
00FF	...
0100	9A
0101	23
0102	8A
0103	43
0104	20
0105	84
0106	44
0107	20
0108	...

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 130

Altro Esempio

Sequenza fetch/execute:

- IP = 0103 Pone il contenuto di IP nell'address bus e la linea R/W ad 1. Carica il byte = 43_{hex} nel registro dati. Incrementa IP (IP=0104). Muove il dato dal registro dati al byte meno significativo di un registro temporaneo T (di 2 byte).
- IP = 0104 Pone il contenuto di IP nell'address bus e la linea R/W ad 1. Carica il byte = 20_{hex} nel registro dati. Incrementa IP (IP=0105). Muove il dato dal registro dati al byte piu' significativo del registro temporaneo T. Muove il contenuto del registro temporaneo nell'address bus e pone la linea R/W ad 1. Carica il byte contenuto nella locazione 2043_{hex} nel registro dati. Somma il dato ad ACC. Setta flag di riporto (carry) se la somma genera overflow.

Lezione 6

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 132

Altro Esempio

Sequenza fetch/execute:

- IP = 0105 Pone il contenuto di IP nell'address bus e la linea R/W ad 1.
Carica il codice operativo 84_{hex} nel registro dati. Incrementa IP (IP=0106). Muove il dato dal registro dati ad IR. Decodifica il codice operativo (la decodifica segnala che i prossimi 2 byte in memoria contengono l'indirizzo del dato da utilizzare).
- IP = 0106 Pone il contenuto di IP nell'address bus e la linea R/W ad 1.
Carica il byte = 44_{hex} nel registro dati. Incrementa IP (IP=0107).
Muove il dato dal registro dati al byte meno significativo del registro temporaneo T.
- IP = 0107 Pone il contenuto di IP nell'address bus e la linea R/W ad 1.
Carica il byte = 20_{hex} nel registro dati. Incrementa IP (IP=0108).
Muove il dato dal registro dati al byte più significativo del registro temporaneo. Muove il contenuto del registro temporaneo.
Muove il dato da ACC al registro dati e pone R/W a 0 (scrittura).