

SISTEMI A MICROPROCESSORE E MICROCONTROLLORE

CONCETTI INTRODUTTIVI

Memorie

Le memorie sono dei sistemi digitali per la *registrazione delle informazioni*. Sono costituite da *registri di memoria accatastati*. L'espansione della capacità dei calcolatori e le soluzioni tecnologiche per l'integrazione dei dispositivi sempre più innovative hanno spinto a costruire memorie con maggiore *capacità* di accumulo, sia per numero di registri, sia per dimensione di ogni singolo registro. Per semplicità ci riferiremo a schemi di principio di memorie aventi registri da 8 bit (1 byte) e un numero limitato di byte.



Dispositivi di base

L'elemento base dei sistemi di memorizzazione digitale è un dispositivo elementare di memoria, noto col nome di **bistabile o flip-flop**.

Questo dispositivo è in grado di mantenere l'uscita anche dopo la rimozione del segnale di ingresso che la comanda, grazie a un meccanismo interno di **autosostenimento** dell'informazione.

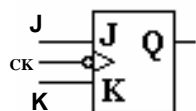
La cella elementare memorizza l'informazione elementare, detta bit.

Il bit è la cifra binaria, che può valere 0 o 1, valori simbolici corrispondenti rispettivamente ai due stati elettrici di assenza e presenza di tensione:

assenza di tensione = 0 V = livello L (Low) = '0'
presenza di tensione = 5 V = livello H (High) = '1'

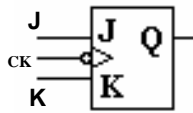
3

FLIP FLOP J K



CK	J	K	Q(t+1)	
↓	0	0	Q(t)	MEM
↓	0	1	0	RESET
↓	1	0	1	SET
↓	1	1	$\overline{Q(t)}$	TOGGLE

4



Il flip-flop è un elemento **sequenziale**, in quanto il valore futuro dell'uscita $Q(t + 1)$ all'istante $t + 1$ dipende, oltre che dagli ingressi J e K al medesimo istante, anche dal valore $Q(t)$ dell'uscita all'istante precedente.

Nella versione **sincrona** il passaggio da $Q(t)$ a $Q(t + 1)$ avviene a intervalli discreti in sincronismo con l'evento di **clock CK**. Il clock è rappresentato nella tabella di verità con il simbolo \downarrow ; la freccia diretta verso il basso esprime il concetto che l'azione del clock ha luogo in coincidenza con la transizione del segnale da livello alto a livello basso. Si dice che il sistema è sensibile al fronte di discesa del clock. **NB:** Esiste anche la possibilità opposta.

CK	J	K	Q(t+1)	
\downarrow	0	0	$Q(t)$	MEM
\downarrow	0	1	0	RESET
\downarrow	1	0	1	SET
\downarrow	1	1	$\overline{Q(t)}$	TOGGLE

Dalla tabella si vede che:

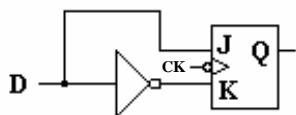
se $JK = 00$ risulta $Q(t+1) = Q(t)$, cioè il dispositivo conserva il valore passato dell'uscita, ovvero si trova in una condizione di memoria (MEM);

se $JK=01$ l'uscita futura assume valore 0

mentre se $JK = 10$ assume valore 1, indipendentemente dal suo valore passato: queste configurazioni di ingresso determinano quindi la scrittura rispettivamente del bit 0 (RESET) e 1 (SET) all'uscita del bistabile;

la quarta combinazione determina la complementazione del valore passato, ma è inutilizzata nei sistemi di sola memorizzazione.

Da queste osservazioni si può evincere che il bistabile costituisce la cella elementare di memoria, poiché mantiene il dato memorizzato e consente anche di impostarlo, scrivendo '0' o '1' in uscita.



Il tipo D si ricava dal JK collegando l'ingresso D a J e a K nel modo a lato. Si hanno allora due sole condizioni di funzionamento:

$D = 0 \rightarrow J = 0, K = 1$ reset;
 $D = 1 \rightarrow J = 1, K = 0$ set.

CK	D	$Q(t+1)$	
↓	0	0	RESET
↓	1	1	SET

Possono presentarsi queste tre distinte situazioni:

fronte di discesa del CK assente: condizione di memoria;

fronte di discesa del CK presente:

ingresso $D = 0 \rightarrow Q = 0$ reset

ingresso $D = 1 \rightarrow Q = 1$ set

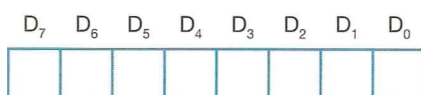
Queste costituiscono l'insieme fondamentale delle operazioni di una cella elementare di memoria, che permette la memorizzazione e la scrittura dei bit '0' e '1'.

7

Registro di memoria

Connettendo più bistabili in cascata si ricava un registro di memoria. I registri sono impiegati nei calcolatori per la memorizzazione temporanea dei dati; possono essere specializzati, cioè inseriti in reticoli circuitali a supporto delle operazioni logiche e matematiche, oppure raggruppati a cascata per formare la sezione di memoria dati.

Tipicamente i bistabili sono raggruppati a gruppi di 8 potendo così accogliere l'informazione base, detta byte.



8

Generalità sulle
memorie

In termini sintetici le memorie possono essere classificate:

RAM (Random Access Memory): Memoria ad accesso casuale. È una memoria volatile di lettura e scrittura; l'accesso al dato avviene istantaneamente, indipendentemente dalla posizione occupata.

ROM (Read Only Memory): Memoria di sola lettura. È una memoria di tipo non volatile, viene incisa alla fonte e non è più modificabile.

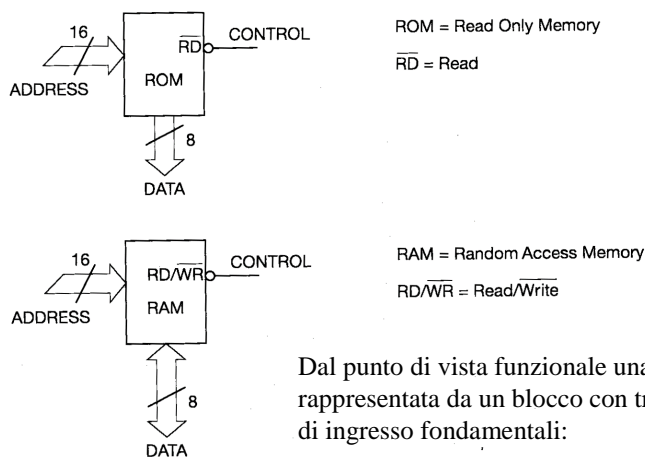
PROM (Programmable ROM): ROM programmabile.

Concettualmente la struttura è come quella della memoria ROM, ma la matrice di dispositivi che registra i dati è programmabile dall'utente.

EPROM (Erasable Programmable ROM): ROM cancellabile e programmabile. Il singolo utente può programmare la matrice desiderata mediante un opportuno programmatore di EPROM e cancellare la programmazione mediante raggi ultravioletti, quindi riprogrammare.

EEPROM (Electrically Erasable Programmable ROM): ROM cancellabile e programmabile elettricamente. A differenza della EPROM non richiede l'esposizione ai raggi ultravioletti per la cancellazione, bensì l'impiego di un più comodo programmatore/cancellatore elettronico.

9



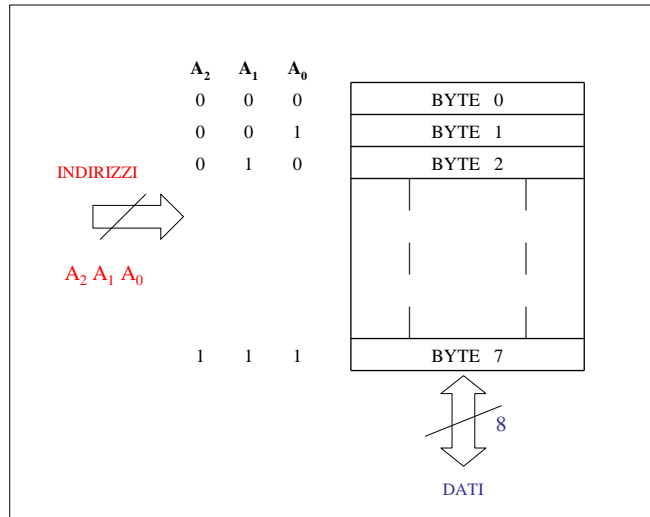
Dal punto di vista funzionale una memoria può essere rappresentata da un blocco con tre gruppi di terminali di ingresso fondamentali:

- DATA: terminali dati;
- ADDRESS: terminali di indirizzo;
- CONTROL: terminali di controllo.

10

Struttura della memoria

Trattiamo la costituzione interna delle memorie, muovendoci attraverso diversi gradi di dettaglio.



**INDIRIZZAMENTO
LOCAZIONE DI
MEMORIA**

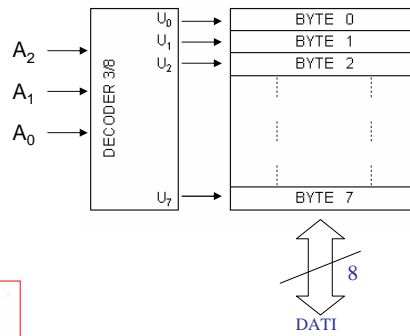
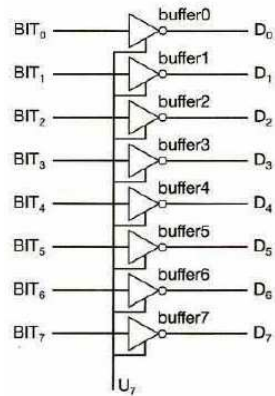


Tabella di verità del DECODER 3/8

A_2	A_1	A_0	U_0	U_1	U_7
0	0	0	1			
0	0	1		0		
.....						
.....			0			
1	1	1				1

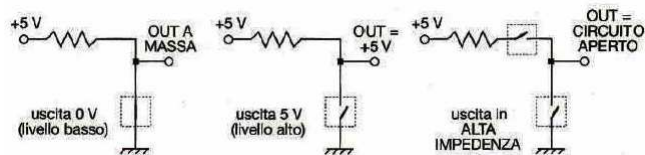
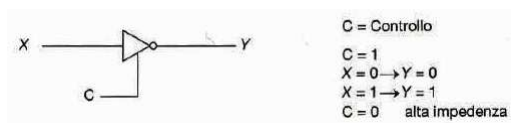
LETTURA BYTE 7



15

II BUFFER

Il Buffer è un dispositivo ad un ingresso, un uscita ed un terminale di controllo C:
con C disabilitato, l'uscita si porta allo stato di alta impedenza o three state.



16

logica cablata e logica programmata

L'avvento dei microprocessori ha determinato una svolta nel progetto dei sistemi logici, decretando il passaggio dalla tradizionale **logica cablata** alla **logica programmata**.

Per **logica cablata** si intende un approccio progettuale di tipo hardware, nel quale un sistema viene progettato assemblando diversi componenti e moduli circuitali, in funzione della applicazione richiesta.

Tali moduli sono circuiti a bassa o media scala di integrazione (SSI = Small Scale Integration, MSI = Medium Scale Integration), il che significa che integrano un numero basso o medio di componenti elettronici.

Per **logica programmata** si intende un approccio progettuale ibrido hardware-software, nel quale:

- ▶ è presente uno strato hardware strutturato in modo permanente;
- ▶ allo strato hardware si sovrappone uno strato software;
- ▶ lo strato software non è rigidamente determinato come quello hardware, ma può essere adattato alle specifiche del progetto.

17

L'integrazione software-hardware dona un elevato grado di flessibilità e semplifica la progettazione dei sistemi, infatti:

- ▶ la funzionalità operativa è dettata dal software, quindi per apportare modifiche al progetto basta modificare il software, senza alterare l'hardware;
- ▶ agendo sul software è semplice correggere il progetto in fase di sviluppo, nonché tenerlo aggiornato;
- ▶ il progettista non deve preoccuparsi dei dettagli costruttivi dello strato hardware.

Nella **logica programmata** i moduli hardware vengono realizzati mediante sistemi LSI e VLSI (LSI = Large Scale Integration; VLSI = Very Large Scale Integration) detti microprocessori e microcontrollori.

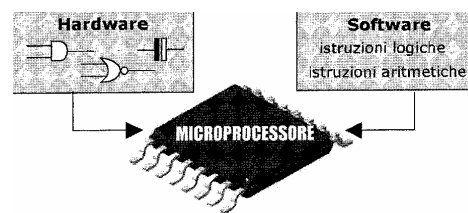
18

microprocessori

Un microprocessore è un dispositivo complesso che integra funzionalità sia hardware che software:

- ▶ dal punto di vista hardware è dotato di circuiti organizzati per le funzionalità interne e per l'interfacciamento con dispositivi esterni;
- ▶ dal punto di vista software è dotato di capacità di calcolo di base logiche e aritmetiche, ma soprattutto può leggere e eseguire programmi.

Un **programma** è una sequenza di istruzioni che comandano al microprocessore l'esecuzione di operazioni elementari.



19

Il **microprocessore** è basilamente dotato di una “intelligenza”, CPU, e una unità logica-aritmetica, ALU, per le operazioni.

Si appoggia a unità esterne, come per esempio memorie di massa, tastiera, video, per scambiare le informazioni da elaborare.

Il **microcontrollore** è un dispositivo completo e autosufficiente;

esso integra entro un unico chip, oltre ai componenti propri del microprocessore, dispositivi aggiuntivi quali memorie, porte di interfacciamento, timer e contatori ...

20

Il microprocessore è:

- ▶ legato al mondo dei computer molto potenti ma anche ingombranti e appesantiti da una serie di sovrastrutture (schermo, tastiera) che ne rallentano il funzionamento;
- ▶ adatto a applicazioni di uso generale come internet e pacchetti applicativi, dove sono necessarie ampie risorse visuali ma non particolari livelli di miniaturizzazione.

21

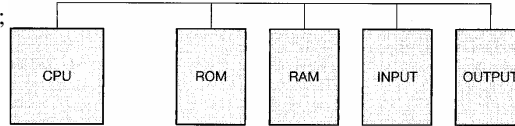
Il microcontrollore :

- ▶ è specializzato, cioè spesso destinato a una sola funzione nel suo ciclo di vita; nella sua memoria interna viene infatti inciso un programma permanente che attraverso i pin svolge funzioni di controllo dei sistemi con i quali è interfacciato;
- ▶ può essere direttamente incorporato (embedded) entro schede elettroniche di tutti i generi, da sistemi robotici a sistemi per la gestione del traffico a giochi, essendo progettato per la massima autosufficienza funzionale;
- ▶ proprio perché incorporato in schede di gestione di processi è progettato per lavorare in tempo reale (real-time), ovvero interagire alla massima velocità con il processo.

22

Basilarmente, di un microcontrollore si possono isolare i seguenti moduli fondamentali:

- 1.CPU;
- 2.sezioni di memoria (RAM, ROM ...);
- 3.sezioni di input-output.



- 1. La CPU = Central Processing Unit**, è il cuore del microcontrollore. Essa:
 - ▶ interpreta ed esegue il programma;
 - ▶ gestisce l'Unità Logico-Aritmetica (ALU = Arithmetic Logic Unit) per l'esecuzione delle istruzioni di programma;
 - ▶ attiva e sincronizza i dispositivi che cooperano all'esecuzione del programma.
- 2. Le sezioni di memoria** sono destinate al mantenimento del programma che la CPU ha in carico e dei dati sui quali la CPU opera. In particolare si distinguono:
 - ▶ memoria ROM dove sono incisi dati strutturali dell'elaboratore;
 - ▶ memoria RAM dove è memorizzato il programma che la CPU deve eseguire e i dati sui quali deve operare.
- 3. Le sezioni di input-output** fungono da interfaccia tra la CPU e le unità esterne al microcontrollore. In particolare si distinguono:
 - ▶ porte di INPUT, ad esempio per la lettura dello stato di interruttori;
 - ▶ porte di OUTPUT, ad esempio per il comando di led e relè.

23

Descrizione dei BUS

I dati che circolano in un microcontrollore sono di tipo binario. Essi transitano in parallelo su gruppi di linee digitali, delle quali ognuna trasporta un bit.

Un gruppo di linee è denominato **BUS**, a indicare il fatto che a esso è affidato il trasporto contemporaneo di più dati.

- ▶ la lettera identifica il tipo BUS;
- ▶ il numero identifica la posizione, a partire dal numero 0.

24

Collegamento a BUS

La CPU è interconnessa con innumerevoli dispositivi. Un collegamento punto a punto tra la CPU e ciascun dispositivo comporterebbe una proliferazione di linee digitali.

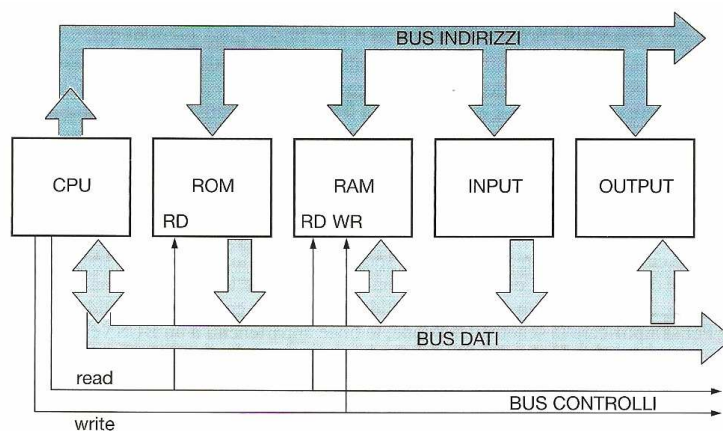
Per questo viene utilizzata la tecnica del collegamento a BUS. In essa lo stesso gruppo di linee è condiviso sia dalla CPU che da tutti i dispositivi a essa interconnessi secondo la cosiddetta **tecnica della moltiplicazione**. Questa soluzione:

- ▶ semplifica il collegamento riducendo drasticamente il numero di collegamenti:
- ▶ permette in teoria una espansione illimitata, senza bisogno di cablare nuove interconnessioni.

25

In linea di principio, facendo riferimento all'architettura di base, le informazioni canalizzate entro tre distinti BUS:

- BUS DATI;
- BUS INDIRIZZI;
- BUS CONTROLLI.



26

Il **BUS DATI** supporta le informazioni scambiate tra CPU e dispositivi è di tipo **bidirezionale** (convoglia i dati sia dalla CPU ai dispositivi che in senso opposto).

Il **BUS Indirizzi** trasmette in modo **monodirezionale** uscente dalla CPU gli **indirizzi di selezione delle locazioni di memoria** e di selezione del **dispositivo** abilitato al colloquio.

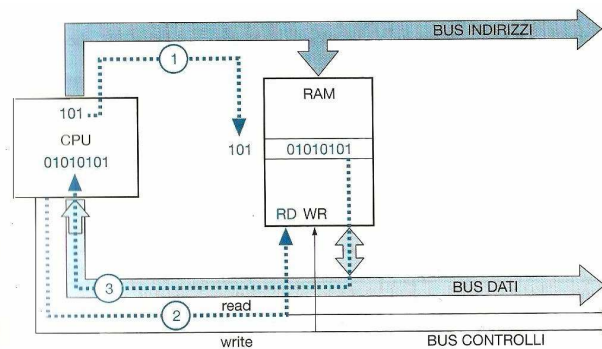
Il **BUS Controlli** è formato da linee specializzate che individualmente attivano delle funzioni nei dispositivi, come la **lettura** (read) e la **scrittura** (write).

27

Come lavorano i BUS

Per osservare la logica e la tempistica della tecnica di comunicazione basata sui bus e il ruolo di ciascun tipo di bus, esemplifichiamo attraverso il processo di lettura di un dato nella memoria, da parte della CPU, che rispetta la successiva sequenza:

- 1- la CPU pone l'**indirizzo** della **locazione di memoria** da leggere (ad esempio 101 = 5) sul BUS INDIRIZZI;
- 2- la CPU invia il segnale di **READ** alla memoria attraverso il BUS CONTROLLI;
- 3- la locazione viene letta e il suo contenuto 0101 0101 viene posto sul BUS DATI. Da qui la CPU lo legge e lo ingloba al suo interno.



28

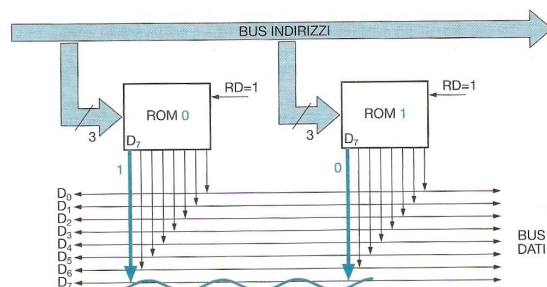
Conflitto di BUS

La tecnica a BUS è essenziale per il risparmio delle linee digitali. Per contro richiede una attenta **disciplina dei collegamenti**, che assicuri che un solo dispositivo alla volta possa colloquiare con la CPU.
Se più dispositivi accedono al BUS contemporaneamente infatti, si genera un **conflitto di bus**.

29

Esempio: Colloquio con due memorie ROM

Entrambe le memorie ROM 0 e ROM 1 sono abilitate alla lettura dal segnale $RD = 1$ ma esse hanno uscita differente. L'uscita D7 di ROM 1 infatti vale '1' mentre quella di ROM 0 vale '0'. Sulla linea D7 pertanto accedono due valori digitali opposti, quindi non è presente alcuna informazione significativa. Anzi, dal un punto di vista elettrico si stabilisce un cortocircuito franco tra le due uscite attraverso la sezione di linea D7 interposta tra esse, perché il livello 1 è una tensione di 5 V mentre il livello 0 è la massa.



30

Abilitazione dispositivi

Il conflitto di bus viene risolto grazie all'attività di governo della CPU, che gestisce opportunamente l'abilitazione dei dispositivi.

L'abilitazione viene comandata attraverso il terminale di selezione dispositivo (Chip Select).

In ragione del valore di CS, il dispositivo si comporta in questi due modi:

- se $CS = 0$ la logica three-state all'uscita viene posta in condizione buffer con circuito aperto, pertanto le uscite risultano virtualmente scollegate;
- se $CS = 1$ le uscite sono libere di porre i valori digitali 0 o 1 sui terminali che fanno capo al bus.

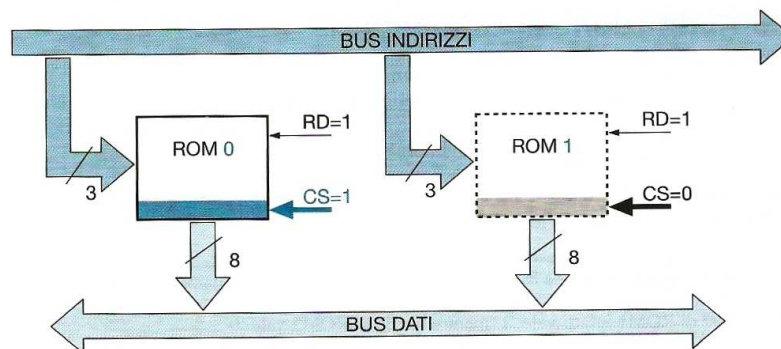
La CPU, nella sua funzione di arbitro, porta alto solo uno dei terminali CS, quindi abilita al colloquio un solo dispositivo; i dispositivi che ricevono $CS = 0$ risultano virtualmente scollegati.

31

Abilitazione memoria ROM.

La figura seguente spiega come la CPU operi per impedire un conflitto di BUS.

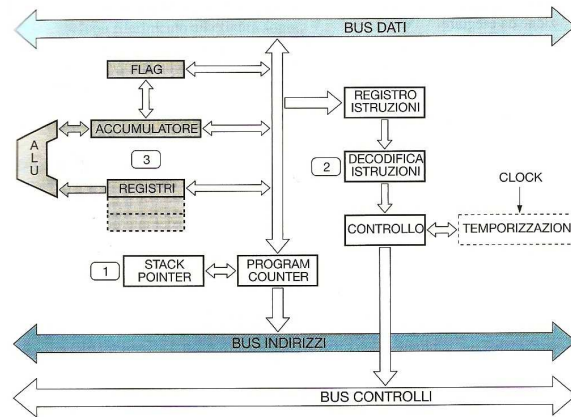
Essa abilita solo ROM0 al colloquio, ponendo a 1 il suo terminale CS e mantenendo a 0 quello di ROM1. Di conseguenza i dati di ROM0 pervengono al bus dati, mentre le uscite di ROM1 rimangono virtualmente disconnesse.



32

Architettura CPU

Non esiste un modello univoco di CPU in quanto ogni famiglia di microprocessori ha struttura hardware e soluzioni software diverse; inoltre con l'evoluzione dei microprocessori, le CPU hanno nel tempo assunto architetture sempre più complesse allo scopo di ottimizzarne l'efficacia e la velocità. Esistono però alcuni tratti ricorrenti.



33

Da un punto di vista concettuale possiamo ascrivere i registri e dispositivi a tre categorie:

1. Registri deputati al prelievo delle istruzioni:

Program Counter e **Stack Pointer**, che servono a individuare la posizione dell'istruzione nella memoria;

2. Registri e dispositivi deputati al riconoscimento delle istruzioni:

Registro Istruzioni, **Decodifica Istruzioni**, **Controllo**, **Temporizzazione** che servono a interpretare l'istruzione e diramare gli ordini agli apparati esecutivi;

3. Registri e dispositivi deputati alla esecuzione delle istruzioni:

Accumulatore, **Registri**, **ALU**, **Flag**, che servono a mettere in atto fattivamente le operazioni comandate dall'istruzione.

34

1. Registri deputati al prelievo delle istruzioni

PROGRAM COUNTER (PC): indica a quale indirizzo di memoria prelevare l'istruzione ancora da eseguire. Una volta eseguita l'istruzione, accade che:

- il PC viene incrementato, per puntare alla successiva istruzione;
- nel caso si debba eseguire un salto, il PC viene forzato al nuovo indirizzo cui saltare.

STACK POINTER (SP): in esso la CPU deposita l'indirizzo di ritorno, nel caso di salti causati da chiamate a sottoprogrammi. In particolare:

- l'indirizzo dell'istruzione corrente viene caricato dal PC entro lo SP;
- al ritorno dalla chiamata a sottoprogramma SP ripristina l'indirizzo in PC;
- in caso di chiamate nidificate, gli indirizzi vengono accatastati (stack significa appunto catasta);
- la modalità di salvataggio e ripescaggio degli indirizzi accatastati è del tipo LIFO (Last In First out = ultimo a entrare, primo a uscire).

35

2. Registri e dispositivi deputati al riconoscimento delle istruzioni

REGISTRO ISTRUZIONI: in esso viene depositato il codice operativo dell'istruzione da eseguire.

DECODIFICA ISTRUZIONI: qui la CPU compara l'istruzione con una lista interna che contiene il set di istruzioni, ovvero l'insieme delle istruzioni che la CPU riconosce. Nella lista, per ogni istruzione, è registrato un microprogramma di implementazione, che viene eseguito.

CONTROLLO: in questa sezione vengono generati i segnali destinati al bus controlli, necessari per completare la lettura dell'istruzione.

TEMPORIZZAZIONE: questa sezione genera sequenze predefinite di impulsi, con un ritmo dettato dal segnale di clock, per comandare elettronicamente i dispositivi coinvolti nell'istruzione.

CLOCK: è un segnale a onda quadra che ritma l'operato del microcontrollore.

36

3. Registri e dispositivi deputati alla esecuzione delle istruzioni

ACCUMULATORE: è il principale registro di memoria, innervato con tutti gli altri registri, infatti:

- quasi tutte le istruzioni lo utilizzano come registro per i dati in ingresso;
- quasi tutte le istruzioni depositano in esso il risultato.

FLAG: è un registro che segnala (flag = bandiera) mediante ogni singolo bit delle particolari condizioni; per esempio:

- il flag di zero è un bit che segnala, mediante il suo stato 1 o 0, se una operazione ha dato risultato uguale o diverso da zero;
- il flag di carry è un bit che registra il valore del riporto generato da una somma.

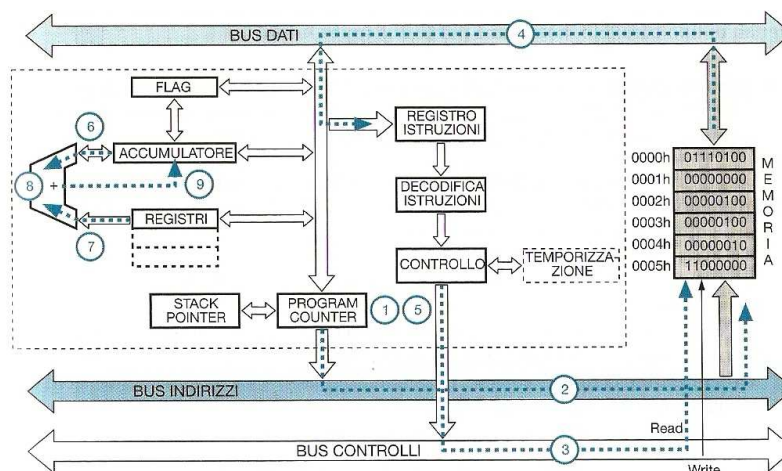
ALU (Arithmetic Logic Unit) : esegue operazioni aritmetiche e logiche.

REGISTRI: sono registri di memoria utilizzati per depositare dati in modo temporaneo, nel corso dell'esecuzione delle operazioni.

37

Per comprendere come cooperano i vari elementi della CPU è necessario innanzitutto sapere che il microprocessore passa di continuo attraverso due fasi operative:

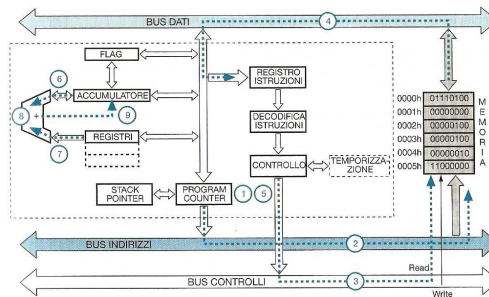
- fase di **Fetch**: in essa la CPU acquisisce dalla memoria l'istruzione (fetch = ricerca);
- fase di **Execute**: in essa la CPU esegue l'istruzione.



38

La **fase di fetch** contempla i passi seguenti:

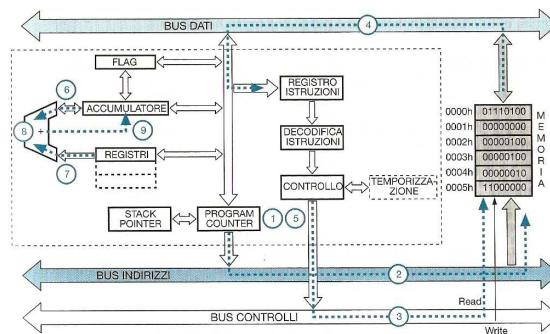
1. Nel Program Counter viene impostato l'indirizzo dell'istruzione da reperire in memoria. In particolare dopo il reset il PC vale solitamente 0000h, cioè punta alla cima della memoria.
2. Il contenuto del PC viene inviato sul Bus Indirizzi, diretto a selezionare la locazione di memoria.
3. La CPU attiva il controllo Read per comandare una operazione di lettura in memoria.
4. La CPU legge la parola di memoria attraverso il Bus Dati e la deposita nel Registro Istruzioni.
5. L'unità di decodifica riconosce il codice operativo presente nel Registro Istruzioni.



Supponiamo che l'istruzione sia **ADD A, B** che significa "somma l'accumulatore A al registro temporaneo B".

In tal caso la **fase di execute** contempla i seguenti passi successivi.

6. La CPU trasferisce il contenuto dell'accumulatore nella ALU.
7. La CPU trasferisce il contenuto del registro temporaneo B nella ALU.
8. La CPU comanda alla ALU di effettuare la somma.
9. La CPU comanda il trasferimento del risultato nell'accumulatore; il dato precedente è sovrascritto



Successivamente, non essendo presenti salti in questo esempio, la CPU incrementa il PC, per predisporre la lettura della locazione successiva di memoria, dando inizio alla nuova fase di fetch.

Qualora una istruzione sia composta da più parole di memoria, la fase di fetch prevede più cicli di lettura in sequenza per il reperimento degli operandi.