

COMPLEMENTI terzo modulo

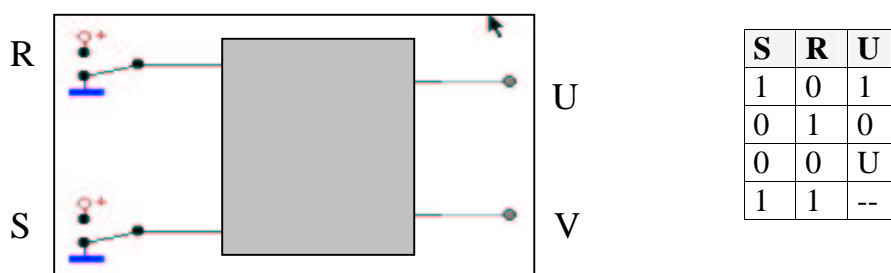
Lezione 1

Il flip-flop

Parlando di macchine elettriche o di reti logiche abbiamo sempre dato per scontato che l'effetto della pressione di un pulsante fosse contemporaneo alla pressione stessa e che esso cessasse al cessare della pressione.

Tutto ciò non si realizza quando utilizziamo una particolare rete logica detta flip-flop.

Possiamo inizialmente immaginarla come una scatola nera con due fili in entrata e due in uscita.



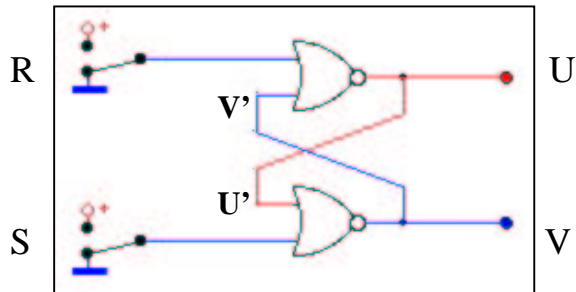
I due tasti S e R si chiamano rispettivamente Set e Reset. Se uno dei due viene tenuto premuto per un tempo opportuno (comunque piccolissimo) la tensione all'uscita assume un determinato valore (vedi tabella) e mantiene tale tensione anche dopo che il tasto è stato rilasciato. In uscita U rappresenta il valore memorizzato, V il suo complemento.

Dalla tabella osserviamo che:

- i due tasti non possono essere premuti contemporaneamente, quindi lo stato $S=1$ ed $R=1$ non è ammesso;
- se premiamo S ($S=1$) in uscita avremo 1;
- se premiamo R ($R=1$) in uscita ci sarà 0;
- infine, cosa ben più importante, se ambedue i tasti vengono rilasciati ($S=0$, $R=0$) lo stato dell'uscita è quello determinato dall'ultimo tasto premuto.

Ciò vuol dire che l'ultimo impulso inviato è memorizzato nell'uscita U fino alla prossima pressione di uno dei due tasti e quindi questa misteriosa rete memorizza un bit di informazione. Negli attuali circuiti digitali questo dispositivo costituisce una unità di memoria.

Vediamo ora l'interno della scatola misteriosa.



La configurazione rappresentata è una configurazione stabile in cui $S=0$ ed $R=0$; il livello di tensione alto (1) è rappresentato dal colore rosso; quello basso (0) dal colore blu.

Osserviamo che l'incrocio delle entrate e delle uscite genera una retroazione che rende stabile il livello di tensione che si è instaurato in uscita.

Per capire meglio il funzionamento possiamo costruire la seguente tabella, dove sono evidenziate le configurazioni stabili ($U'=U$ e $V'=V$):

S	R	U'	V'	U	V
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0

Una visione dinamica del processo di memorizzazione di un bit è offerta dall'applicazione per il progetto e la simulazione di reti logiche disponibile attraverso le pagine ipertestuali di questa unità.

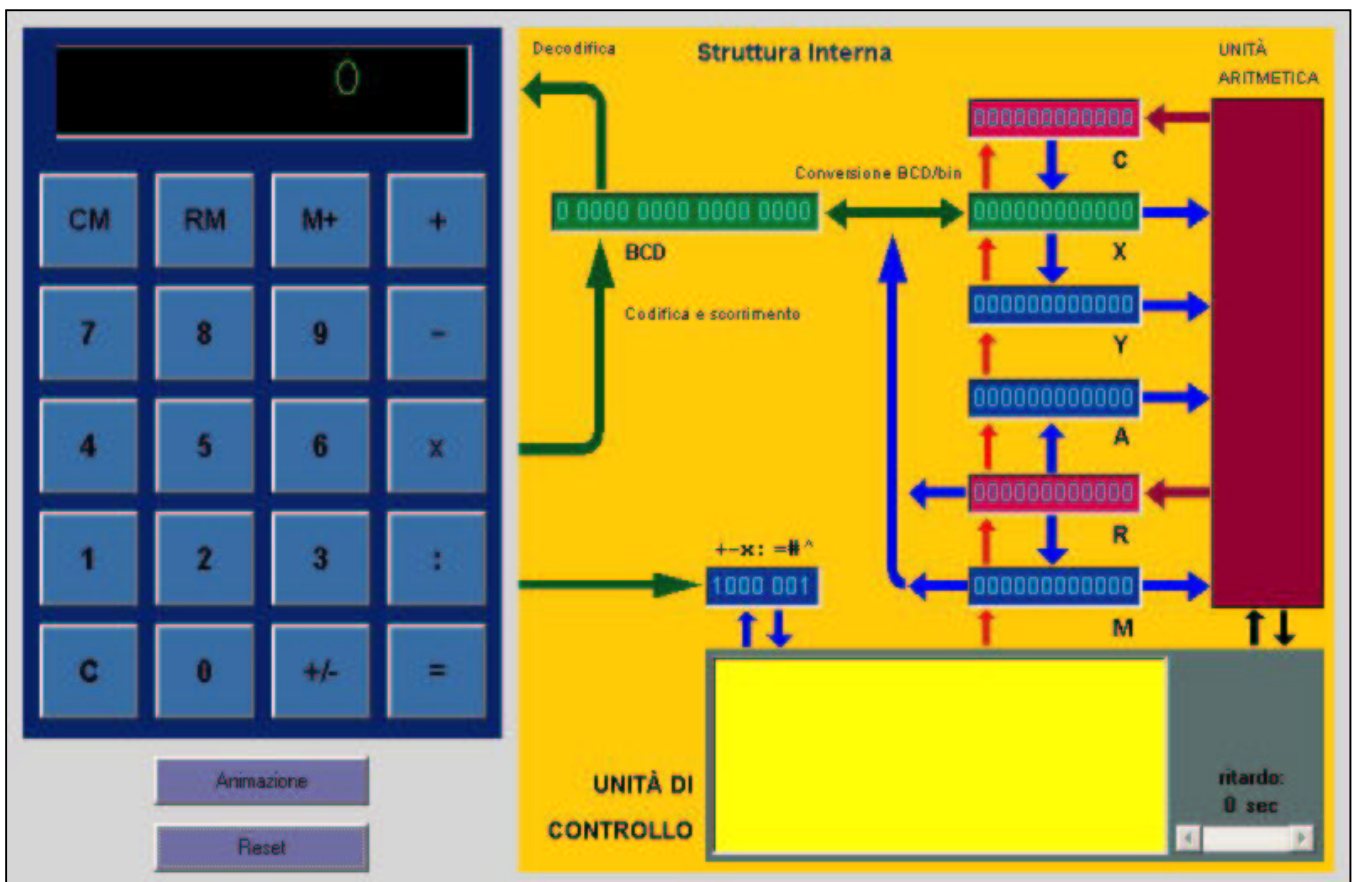
Lezione 2

All'interno di un circuito integrato

Abbiamo visto come si possano memorizzare e recuperare le informazioni per mezzo di circuiti logici. Se vogliamo conoscere come queste informazioni vengono elaborate al fine di far funzionare la nostra calcolatrice tascabile, dobbiamo studiare quali sono i vari blocchi funzionali realizzati all'interno di un circuito integrato e come essi riescono a comunicare fra loro e operare insieme.

Fatta questa premessa osserviamo nella tavola seguente i principali sottosistemi di una calcolatrice e le connessioni (bus) attraverso le quali le informazioni vengono trasmesse da un sottosistema all'altro sotto forma di segnali elettrici.

Con *sottosistema* indichiamo un sistema più piccolo posto all'interno di uno più grande. Le grosse frecce rappresentano possibili percorsi dei dati fra i vari sottosistemi: ognuno di essi è in realtà costituito da diversi conduttori sui quali transitano simultaneamente i segnali elettrici.



Oltre al codificatore e al decodificatore già citati in precedenza, vi compaiono:

- **L'Unità di Controllo:** sottosistema che controlla tutte le altre componenti attivando e disattivando in modo opportuno le varie reti logiche e i registri di memorizzazione. Le modalità secondo cui vengono fatte queste operazioni dipendono dalle istruzioni del microprogramma che l'unità di controllo legge, decodifica ed esegue.
- **La memoria a microprogramma:** vi sono conservate stabilmente le istruzioni, organizzate in *routine*, che realizzano le operazioni. Visto che il microprogramma è memorizzato in modo permanente e non modificabile, il supporto di questo sistema è una memoria di tipo ROM: Read Only Memory (nella figura non è stato isolato questo sottosistema ma l'UC "visualizza" le istruzioni eseguite).
- **L'Unità Aritmetica:** componente elettronica in grado di eseguire le operazioni dell'aritmetica eventualmente eseguendo le routine microprogrammate nella ROM (troveremo nel computer la generalizzazione di questa componente con il nome di ALU: Unità Aritmetico -Logica).

Vi sono poi alcuni registri per la memorizzazione temporanea dei dati fra i quali i più importanti sono:

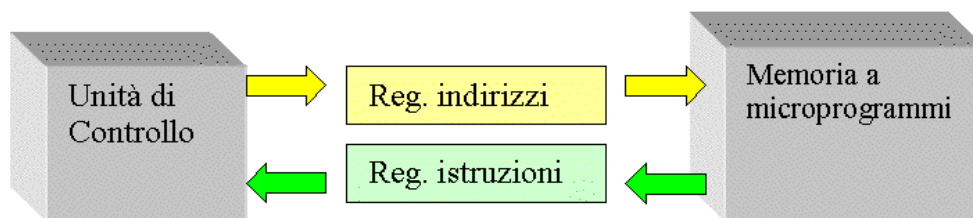
- il **registro X** (collegato al registro BCD e quindi al codificatore e al decodificatore) che memorizza i dati numerici provenienti dalla tastiera e/o visualizzati sul display. Visto da questo punto di vista il registro X ha in un certo senso la funzione che nel computer avrà la memoria video.
- Il **registro Y** serve per conservare il secondo operando; riceve una copia del valore che c'è in X quando si digita un operatore.
- Il **registro accumulatore A** per conservare i risultati intermedi; in genere svolge anche la funzione di primo operando.
- Il **registro R** per contenere temporaneamente il risultato di una operazione dell'Unità Aritmetica (UA).
- Il **registro flag** i cui bit, posti a 0 o a 1, rappresentano lo stato della macchina e cioè vengono posti a 0 o a 1 a seconda dell'operazione che si vuole svolgere, se c'è già una operazione in sospeso o se c'è già un dato numerico. All'accensione della calcolatrice esso pone a 1 il bit della addizione.
- Il **registro C** per contenere temporaneamente il risultato del cambio di segno.

Come vengono controllati i sottosistemi.

La prima cosa che dobbiamo capire è che tutti i sottosistemi sono direttamente connessi al sottosistema *Unità di Controllo* tramite una rete di conduttori elettrici, non tutti illustrati nella figura.

Compito dell'Unità di Controllo è di indicare ad ogni sottosistema quando deve agire e che cosa deve fare. L'Unità di Controllo opera interpretando le istruzioni che legge, una alla volta, dalla unità di memoria in cui esse sono memorizzate in modo permanente all'interno del chip; questa componente è chiamata *Memoria a microprogramma*.

Ciascuna istruzione è stata permanentemente memorizzata in una particolare locazione, ad un particolare indirizzo della Memoria a microprogramma. Ogni locazione di memoria, e quindi l'istruzione che essa contiene, è identificata da un numero detto indirizzo, proprio come un numero civico di casa, o un numero di telefono. L'Unità di controllo accede in successione a ciascuna istruzione depositando l'indirizzo relativo in una unità di memoria temporanea chiamata Registro di indirizzamento. In seguito, viene attivata la locazione indirizzata, si reperisce l'istruzione e se ne trasferisce immediatamente una copia in un'altra unità di memoria temporanea chiamata Registro delle istruzioni, perché possa venire decodificata ed eseguita dall'Unità di controllo.



Quando accendiamo la calcolatrice, automaticamente l'unità di controllo legge, attraverso la procedura che abbiamo visto, l'istruzione corrispondente alla prima locazione che stabilisce di inizializzare tutte le informazioni presenti nei registri, in particolare azzerando i registri destinati ai dati numerici.

Tutto avviene nel tempo di un ciclo di istruzione mediante un segnale di controllo inviato a tutti i registri che, come detto sopra, sono memorie temporanee per numeri ed altri tipi di informazione.

Nel successivo ciclo di istruzione l'unità di controllo inizia quella che viene chiamata la routine di attesa, eseguendo una istruzione che, tradotta nel linguaggio comune suonerebbe: "Verifica se ci sono segnali provenienti da tastiera. Se non riconosci alcun segnale in arrivo ripeti questa stessa istruzione. Se trovi un segnale, passa alla istruzione successiva"; cioè l'unità aspetta che un tasto qualsiasi venga premuto. Questa routine viene rieseguita anche al termine di qualunque routine che l'unità ha eseguito nello svolgimento delle operazioni richieste.

Il sottosistema per la decodifica attraverso il display opera contemporaneamente. Il suo compito è di mantenere illuminato il display, visualizzare le cifre che al momento sono rappresentate nel registro del display (**BCD**), attivare le opportune linee affinché i pixel ricevano corrente ai tempi dovuti. Esso in questo caso visualizza uno 0 (che sarebbe seguito dal punto decimale nelle calcolatrici normali) che è quanto in pratica vediamo prima di iniziare una operazione. (nella nostra simulazione il registro display è un circuito di memorizzazione temporanea di un numero intero a 4 cifre, più l'eventuale segno meno)

Vogliamo ora seguire il percorso dei dati durante l'esecuzione di una operazione o di una sequenza di operazioni.

Il comportamento descritto qui di seguito riguarda la simulazione presente nella applet, ma ognuno potrà constatare che esso è alquanto simile se non uguale a quello della gran parte delle calcolatrici non scientifiche attualmente in uso.

Supponiamo ora di voler calcolare 4×3 .

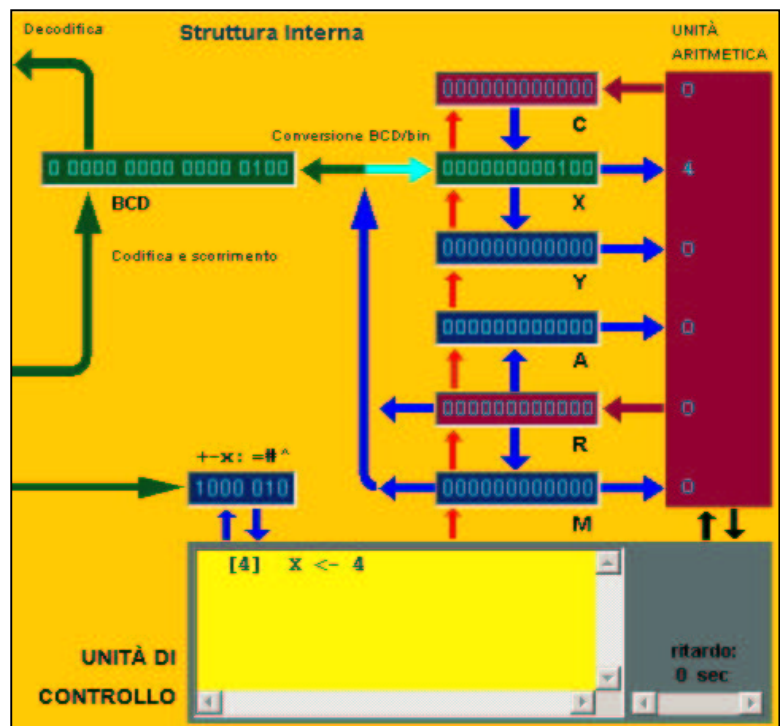
Premiamo il tasto 4

Premiamo ora il tasto 4. Viene trasmesso un segnale su una particolare linea di ingresso dal tasto al sottosistema codificatore di tastiera che abbiamo analizzato nel precedente modulo.

Tale codificatore genera il numero "4" secondo il codice BCD che viene poi convertito in binario e trasmesso al registro X.

Il codificatore invia pure un segnale all'unità di controllo, avvertendola che è stato premuto un tasto numerico. In questo momento, nel registro flag è attivo il bit relativo all'operatore +; l'unità di controllo porta a 1 anche il bit che serve a rammentargli che è stato premuto il primo tasto per l'esecuzione di un nuovo calcolo, poi invia un segnale al registro display affinché traduca il codice in segnali per il decodificatore. Poiché questo registro possiede posti separati di deposito (nel nostro caso per quattro cifre, nelle calcolatrici usuali per otto) il registro provvede a collocare il "4" nella posizione più a destra salvo poi farlo slittare verso sinistra se vengono premuti ulteriori tasti numerici.

Tasto	on	4
registro X	0	4
registro Y	0	0
registro A	0	0
registro R	0	0
flag	+	+



Dopo numerosi cicli di istruzione, in cui l'unità di controllo ha eseguito metodicamente le istruzioni e deciso qual è la successiva istruzione, il nostro "4" è stato visualizzato e l'unità di controllo può tornare alla sua routine di attesa. Tutto ciò ha richiesto una frazione impercettibile di secondo. Si

scopre così quante diverse cose devono venire svolte, in un sistema digitale, per eseguire un compito anche molto semplice, e come tutto avvenga in tempi assai rapidi.

Si può dire che è proprio la rapidità il vero segreto del successo dell'elettronica digitale. Ogni lavoro viene scomposto in tanti piccoli passi che possono essere interpretati e/o elaborati da circuiti elettronici relativamente semplici.

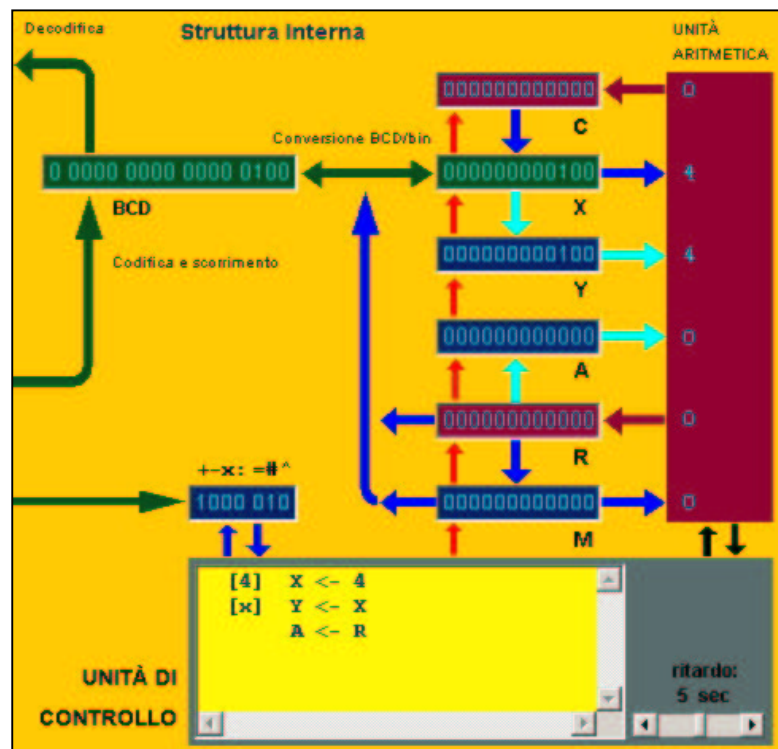
Premiamo il tasto ×

Quando premiamo il tasto relativo ad un operatore, il codificatore indica all'unità di controllo che ciò è avvenuto.

Riconosciuto che questa volta è stato ricevuto un comando di moltiplicazione, anziché un segnale relativo ad un numero, l'unità di controllo verifica nel registro flag se risultano indicazioni di tasti operativi premuti in precedenza. Visto che supponiamo di aver appena acceso la calcolatrice, nel registro flag è attivato il bit relativo all'addizione. Ciò porta l'unità di controllo ad una nuova istruzione, la prima di una sequenza di istruzioni del programma per effettuare l'operazione in sospeso.

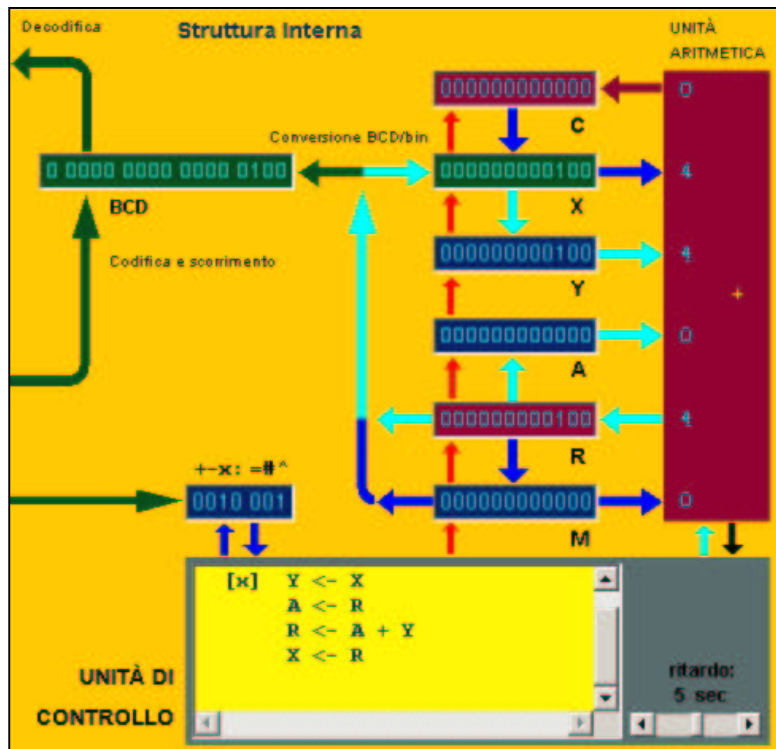
Questa sequenza si compone di tre fasi: per prima cosa il contenuto del registro X (4) viene ricopiato nel registro Y e simultaneamente una copia del contenuto del registro R (0) viene registrata in A;

Tasto	on	4	x
registro X	0	4	
registro Y	0	0	4
registro A	0	0	0
registro R	0	0	
flag	+	+	×



poi l'unità di controllo fa trasferire il contenuto dei due registri Y e A nelle reti logiche dell'Unità Aritmetica alla quale essi sono collegati perché venga effettuato il calcolo richiesto; infine il risultato (4) dell'operazione viene memorizzato nel registro R e da qui ricopiato nel registro X affinché la consueta sequenza di visualizzazione faccia apparire un "4" sul display della calcolatrice.

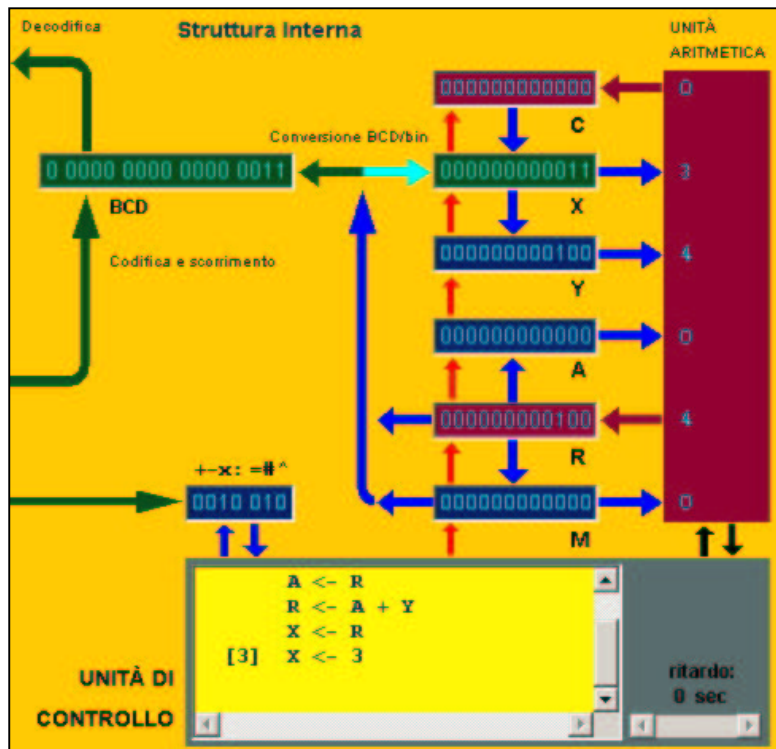
Tasto	on	4	×
registro X	0	4	4
registro Y	0	0	4
registro A	0	0	0
registro R	0	0	4
flag	+	+	×



Anche questa sequenza di operazioni si è svolta in modo rapidissimo e ne troviamo traccia solo nel brevissimo lampeggio del “4” sul display.
Ora l’unità di controllo attiva nel registro flag il bit relativo alla moltiplicazione e riprende la routine di attesa.

Premiamo il tasto 3

Tasto	on	4	×	3
registro X	0	4	4	3
registro Y	0	0	4	4
registro A	0	0	0	0
registro R	0	0	4	4
flag	+	+	×	×

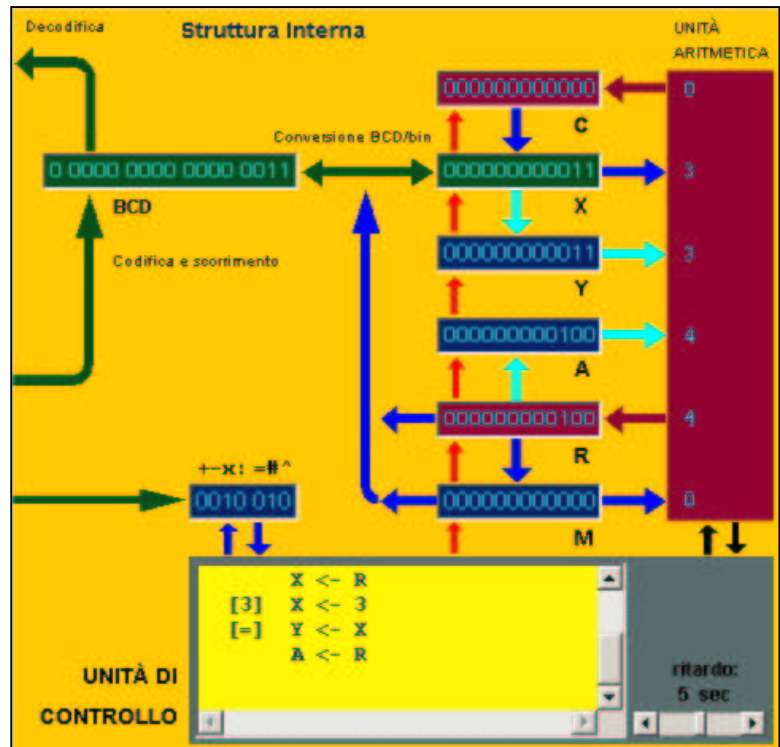


Premendo il tasto “3” si attiva la medesima sequenza operativa già vista per la pressione del tasto “4”. Il registro X viene cancellato e vi viene memorizzato il “3” (subito visualizzato sul display), mentre nel registro flag è memorizzata l’indicazione che è iniziata l’inserzione di un nuovo numero.

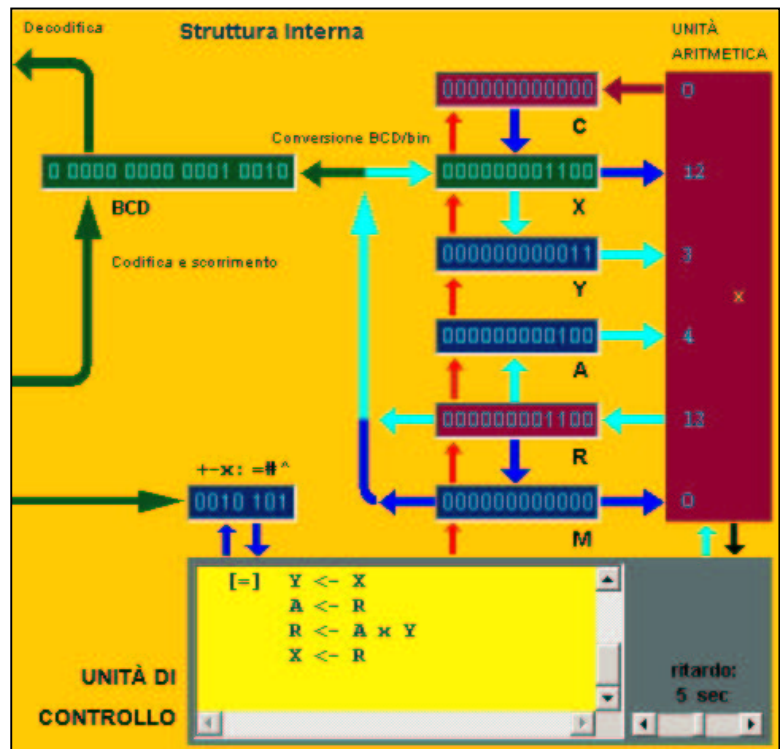
Premiamo il tasto =

La pressione del tasto = viene letta come “esegui il calcolo indicato”. Quindi l’unità di controllo riprende la routine descritta in precedenza.

- Il contenuto di X (3) viene ricopiato in Y e quello di R (4) in A.
- I valori contenuti in Y e in A vengono elaborati dalla UA.



- Il risultato dell’operazione “12” viene memorizzato in R e poi ricopiato in X.
- Dal registro X attraverso il decodificatore viene visualizzato il “12” sul display.



La stessa sequenza verrebbe effettuata anche se al posto dell'uguale si premesse un altro tasto di operazione.

Tasto	on	4	×	3	=
registro X	0	4	4	3	12
registro Y	0	0	4	4	3
registro A	0	0	0	0	4
registro R	0	0	4	4	12
flag	+	+	×	×	×

Casi particolari

- Osserviamo che, alla fine della sequenza di tasti $4 \times 3 =$ non viene eseguito un reset automatico per cui nei vari registri rimangono memorizzati i valori precedenti, in special modo è ancora attivo il bit del flag relativo alla moltiplicazione.
Se viene premuto nuovamente il tasto $=$ l'unità di controllo:
 - fa ricopiare il contenuto di R in A
 - conserva il valore in Y
 - esegue una nuova moltiplicazione per 3.
- La stessa procedura viene attivata anche quando viene premuto il tasto $=$ dopo quello di operazione (es. " $3 \times =$ ") e questa particolarità viene utilizzata per eseguire potenze.
- Se dopo la sequenza " $4 \times 3 =$ " viene introdotto un nuovo numero esso viene memorizzato in X al posto del "12", poi, quando viene premuto il tasto relativo all'operazione, l'unità di controllo azzerà il registro R e riprende l'intera routine del calcolo.
- Se dopo la sequenza " $4 \times 3 =$ " viene premuto il tasto di una nuova operazione (per es. $+$), l'unità di controllo modifica i bit del flag, ripete la routine del calcolo partendo dal trasferimento dei valori presenti in X (12) e in R (12) rispettivamente a Y e ad A e riprende la routine di attesa del secondo operando.

- Se si digita una sequenza del tipo “ $2 \times 3 = 4 = 5 =$ ” il secondo = viene tradotto dall’unità di controllo nella seguente procedura:
 - ricopia il “4” nel registro Y
 - fa eseguire l’operazione segnalata nel flag(“×”)
 - registra in R e X il risultato.

In questo caso il contenuto del registro A non viene mai modificato e ciò consente per esempio di eseguire una serie di conversioni (es. Lire-euro).

- La pressione del tasto M+ corrisponde alla seguente sequenza:
 - il risultato dell’operazione in corso viene addizionato a quello eventualmente già presente in M,
 - R viene azzerato,
 - nel flag si attiva il bit del +.