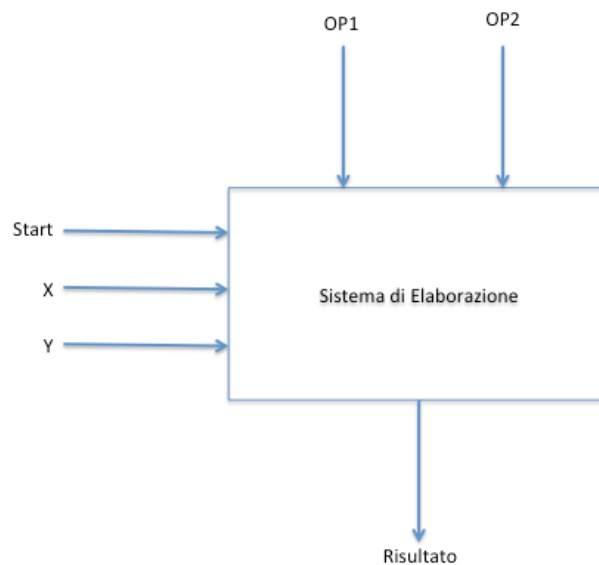


## PROGETTAZIONE DI UNA UNITA' DI CONTROLLO DI UN SISTEMA PER L'ARITMETICA SUI NUMERI NATURALI

---

Si vuole progettare un sistema di elaborazione che implementa le quattro operazioni aritmetiche sui numeri naturali. Il sistema è attivato da un segnale esterno Start. Quando Start=1 allora esso carica dall'esterno i valori Op1 e Op2 degli operandi, e procede al calcolo della operazione specificata dai valori delle variabili X e Y. Gli operandi sono entrambi a 32 bit.



Nel seguito assumiamo che la PO sia data. Si tratta quindi di progettare la UC.

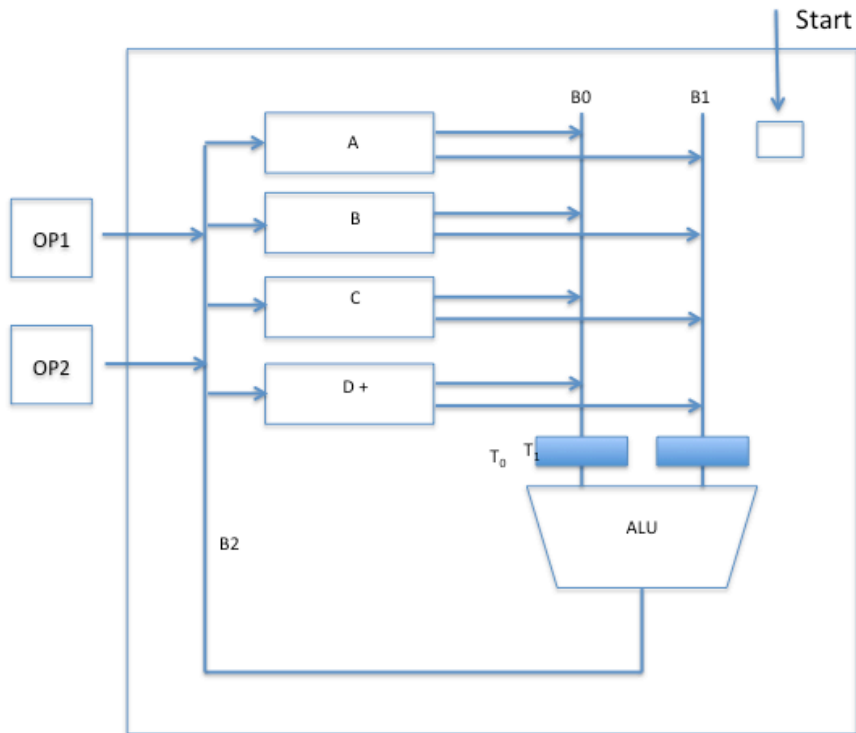
### ARCHITETTURA DELLA PARTE OPERATIVA

---

L'architettura della PO è schematizzata nella seguente figura. Essa comprende:

- quattro registri A, B, C e D collegati in lettura ai bus B0 e B1, ed in scrittura al bus B2 sul quale viaggia l'output dell'ALU. Il registro D è un registro contatore a incremento e decremento. Il registro A è a 64 bit, mentre gli altri sono a 32 bit
- due registri tampone T0 e T1 a 64 bit per stabilizzare gli input dell'ALU
- una ALU a 64 bit che esegue 4 operazioni: (1) la somma ( $\text{output(alu)} = T0 + T1$ ), (2) l'OR ( $\text{output(alu)} = T0 \text{ or } T1$ ), (3) la sottrazione ( $\text{output(alu)} = T0 - T1$ ) e (4) la funzione identità ( $\text{output(alu)} = T1$ )
- un flip flop RS per la memorizzazione del bit Start

I registri sono collegati tramite bus a a 64 bit, come rappresentato in figura.



**Parte Operativa**

## PROGETTAZIONE DELLA UNITÀ DI CONTROLLO

### Microprogramma

Per la stesura del microprogramma assumiamo che i due operandi Op1 e Op2 siano caricati nei registri A e B e che il risultato venga memorizzato nel registro A:

- $A + B \rightarrow A$ ;
- $A * B \rightarrow A$
- $A \text{ div } B \rightarrow A$
- $A - B \rightarrow A$ .

Inoltre, notiamo che l'ALU è in grado di eseguire somme e sottrazioni, ma non moltiplicazioni e divisioni.

- 1) If Start==0  $\Phi$  goto 1
- 2) Op1  $\rightarrow$  A, Op2  $\rightarrow$  B;

```
if X==0 & Y==0 { // A + B  $\rightarrow$  A
  a. A  $\rightarrow$  T0; B  $\rightarrow$  T1;
  b. Alu(T0 + T1)  $\rightarrow$  A; Start=0; goto 1
}
```

```
if X==0 & Y==1 { // A * B  $\rightarrow$  A, B > 0
  // esegui A + A  $\rightarrow$  A B-1 volte; assumiamo B > 0
  c. B  $\rightarrow$  T1;
  d. Alu(T1)  $\rightarrow$  D;
  e. Decr(D); A  $\rightarrow$  T1; //il valore in D è pari a B-1
  f. if D == 0 Start=0; goto 1; //  $\Phi$  è l'istruzione vuota
  g. A  $\rightarrow$  T0;
```

```

h. Alu(T0+T1) → A; Decr(D); goto f;
}

```

```

if X==1 & Y==0 { // A div B → D; A mod B → A
    //esegui A-B → A fino a quando si verifica la condizione A<B
    i. A → T0; B → T1; Azz(D);
    j. if T0 < T1 then D → T1 goto m
    k. Alu(T0-T1) → A; Incr(D);
    l. A → T0 goto j
    m. T1 → A; Start=0; goto 1
}

```

```

if X==1 & Y==1 { // A - B → A
    n. A → T0; B → T1;
    o. Alu(T0 - T1) → A; Start=0; goto 1
}

```

### Segnali di Controllo

I segnali di controllo sono i segnali inviati dalla UC alla PO per determinare il comportamento dei suoi componenti (registri, ALU, ecc.).

#### Registri

- Segnali di abilitazione alla scrittura: A<sub>A</sub>, A<sub>B</sub>, A<sub>C</sub>, A<sub>D</sub>, A<sub>T0</sub>, A<sub>T1</sub>
- Segnali di abilitazione alla lettura:
  - Bus B0: S<sub>A</sub>, S<sub>B</sub>, S<sub>C</sub>, S<sub>D</sub>
  - Bus B1: R<sub>A</sub>, R<sub>B</sub>, R<sub>C</sub>, R<sub>D</sub>

Il registro D è un registro contatore a incremento e decremento, che può essere azzerato e caricato dal bus B2. Il suo funzionamento è regolato dalle seguenti variabili di controllo.

A <sub>D</sub>	K	Z	operazione
0	-	-	Nulla
1	0	0	Carica dal bus B <sub>2</sub>
1	0	1	Azzeramento
1	1	0	Incremento
1	1	1	Decremento

ALU – può eseguire 4 operazioni e, pertanto, necessita di 2 variabili di controllo. Si ipotizza il seguente funzionamento:

L0	L1	Operazione
0	0	T <sub>0</sub> +T <sub>1</sub>
0	1	T <sub>0</sub> -T <sub>1</sub>
1	0	T <sub>0</sub> or t <sub>1</sub>
1	1	T <sub>1</sub>

**Flip flop Start:** segnale Zs di azzeramento.

### Codifica Microistruzioni

Ogni microistruzione che appare in un microprogramma viene codificata assegnando ai segnali di controllo opportuni valori (di seguito si riportano solo alcuni esempi).

microistruz	A <sub>A</sub>	A <sub>B</sub>	A <sub>C</sub>	A <sub>D</sub>	A <sub>T0</sub>	A <sub>T1</sub>	S <sub>A</sub>	S <sub>B</sub>	S <sub>C</sub>	S <sub>D</sub>	R <sub>A</sub>	R <sub>B</sub>	R <sub>C</sub>	R <sub>D</sub>	K	Z	L0	L1	Zs
A → T <sub>0</sub>	0	0	0	0	1	0	1	0	0	0	0	0	0	0	-	-	-	-	0
Alu(T <sub>0</sub> +T <sub>1</sub> )→A	1	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	0	0	0
Incr(D)	0	0	0	1	0	0	-	-	-	-	-	-	-	-	1	-	-	-	0
Φ	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

L'istruzione vuota Φ è una istruzione che non modifica il contenuto dei registri. Essa pertanto si codifica mettendo a 0 i segnali di abilitazione A<sub>x</sub>, dove x ∈ {A, B, C, D}.

### Segnali di Condizione

I segnali di condizione sono inviati dalla PO alla UC. Essi consentono di implementare le istruzioni condizionali presenti nel microprogramma.

Un primo segnale di condizione è il segnale Start che avvia l'esecuzione del microprogramma (microistruzione 1).

Le due istruzioni condizionali (linee f e j) nei microprogrammi delle istruzioni "prodotto" e "divisione"

- if D == 0 Φ goto j;
- if T<sub>0</sub> < T<sub>1</sub> then Φ goto m

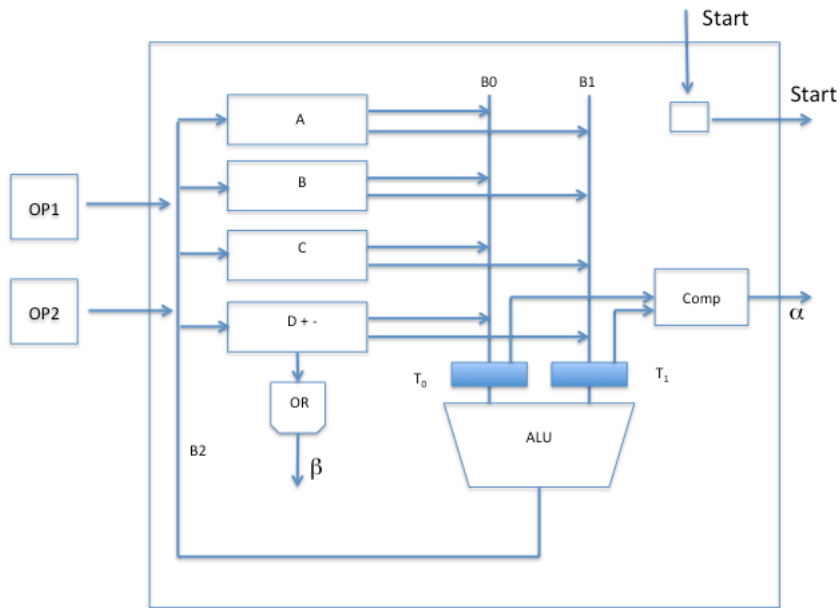
richiedono circuiti aggiuntivi per il controllo delle condizioni. In particolare:

- la condizione D==0 viene testata aggiungendo una porta OR che riceve in ingresso i bit d<sub>1</sub>, ... d<sub>n</sub> del registro D; ovviamente D == 0 sse β = OR(d<sub>1</sub>, ... d<sub>n</sub>) = 0;
- la condizione T<sub>0</sub> < T<sub>1</sub> richiede l'aggiunta di un circuito comparatore che riceve in ingresso i bit dei registri T<sub>0</sub> e T<sub>1</sub> e fornisce in output un segnale α = 1 sse T<sub>0</sub> < T<sub>1</sub>.

Per cui, le sudette istruzioni condizionali possono essere riformulate come segue:

- if β = 0 Φ goto j;
- if α = 1 then Φ goto m

Pertanto, la Parte Operativa genera tre segnali di condizione α, β e Start.



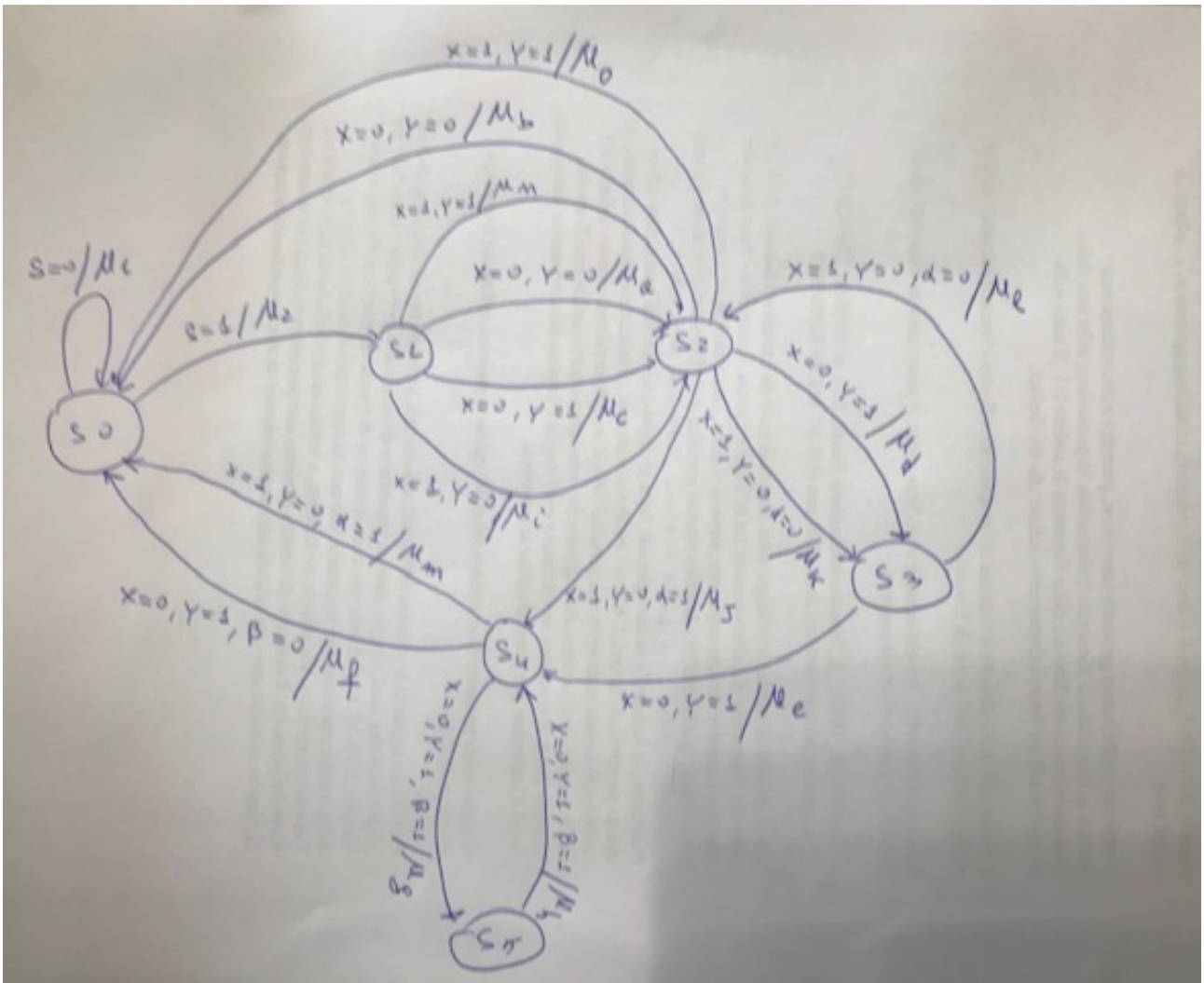
### Unità di Controllo come Automa a Stati Finiti

Il comportamento dell'Unità di Controllo è rappresentato dal seguente automa a stati finiti, dove le etichette I/O degli archi hanno il seguente significato:

- l'input I è una quintupla (start, X, Y,  $\alpha$ ,  $\beta$ ), dove X e Y rappresentano i bit che codificano le 4 istruzioni; per semplicità, in figura sono rappresentati su ogni arco solo i valori dei segnali significativi
- l'output O rappresenta la codifica binaria della microistruzione  $\mu_i$  con etichetta i nel microporgramma.

La corrispondenza tra i valori di X e Y e le istruzioni è la seguente

X Y	operazione
0 0	$A+B \rightarrow A$
0 1	$A*B \rightarrow A$
1 0	$A \text{ div } B \rightarrow A$
1 1	$A-B \rightarrow A$



Come si evince dal grafo, fintantoche  $Start=0$  il sistema rimane nello stato iniziale  $S_0$ . Quando  $Start=1$ , allora il sistema esegue la microistruzione  $\mu_2$  del microprogramma per il caricamento degli operandi e passa nello stato  $S_1$ . Da qui in poi, viene seguito il percorso corrispondente ai valori di  $X$  e  $Y$  che identificano l'istruzione da eseguire.

Per costruire il grafo, a partire dal nodo  $S_1$ , si considera l'istruzione cui è associato il microprogramma più lungo (cioè, col maggior numero di microistruzioni), nella fattispecie quello della moltiplicazione (codice  $X=0, Y=1$ ). A tale istruzione è infatti associato un microprogramma formato da 6 microistruzioni, che si può rappresentare con 5 stati ( $S_1$ - $S_5$ ). Tali stati possono quindi essere parzialmente utilizzati anche dalle altre istruzioni.

Ad esempio, l'istruzione di somma (codice  $X=0, Y=0$ ), che ha un microprogramma formato da 2 microistruzioni, utilizza solo gli stati  $S_1$  e  $S_2$ , mentre la divisione (codice  $X=1, Y=0$ ) utilizza gli stati da  $S_1$  a  $S_4$ .

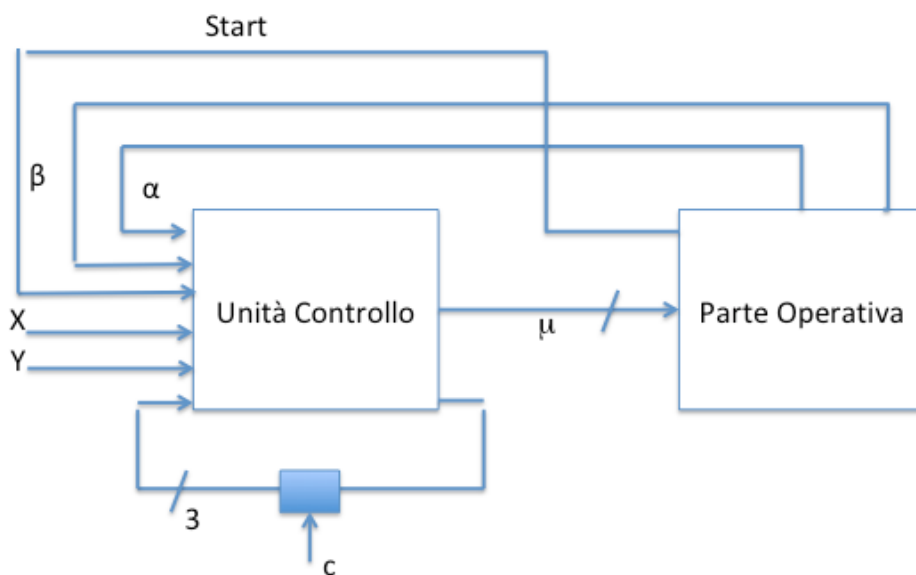
In generale, utilizzando un sottoinsieme degli stati  $S_1$ - $S_5$ , è possibile rappresentare i microprogrammi di tutte le istruzioni.

## SCHEMA ARCHITETTURALE DEL SISTEMA DI ELABORAZIONE

---

Come già menzionato, un sistema di elaborazione consiste di una Parte Operativa ed una Unità di Controllo. L'Unità di Controllo del nostro sistema di elaborazione

- riceve in ingresso i segnali X,Y che indicano l'istruzione da eseguire, più i segnali di condizione  $\alpha$ ,  $\beta$  e Start inviati dalla Parte Operativa
- fornisce in uscita i 20 segnali di controllo  $\mu$  che codificano le microistruzioni.



L'UC è quindi una macchina sequenziale (automa a stati finiti) sincrona con 5 segnali di ingresso (X, Y,  $\alpha$ ,  $\beta$  e start), 3 variabili di anello (necessarie per rappresentare i 6 stati) e 20 segnali di uscita (segnali di controllo della Parte Operativa).

La parte combinatoria dell'UC è una rete con 8 variabili d'ingresso (X, Y,  $\alpha$ ,  $\beta$  e start, più le 3 variabili di anello) e 23 uscite (i 20 segnali di controllo per la Parte Operativa più le 3 variabili di anello).