



Macchine per L'Elaborazione dell'Informazione -

2. Architetture dei Sistemi di Elaborazione

Your Name

Your Title

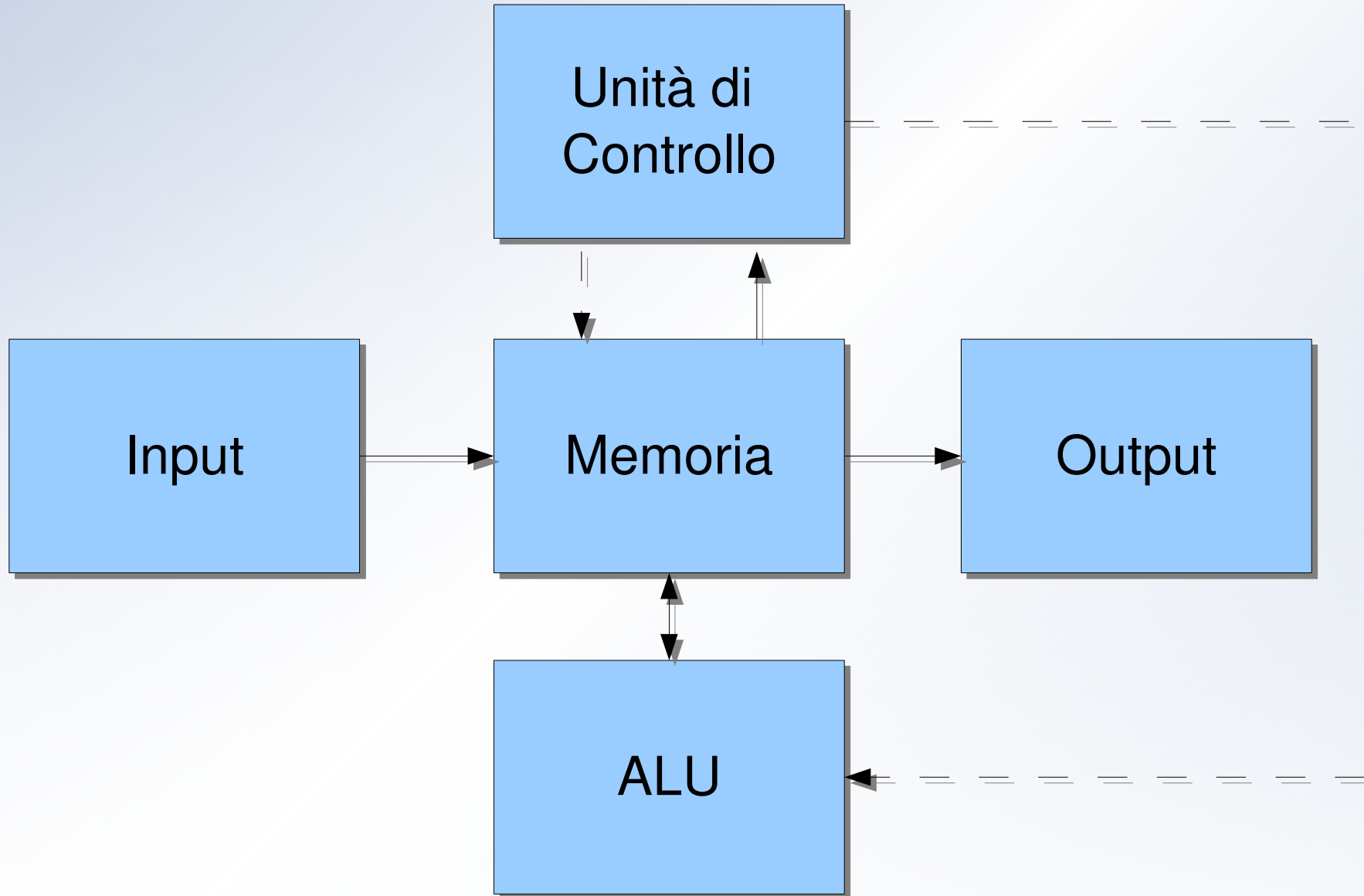
Your Organization (Line #1)

Your Organization (Line #2)

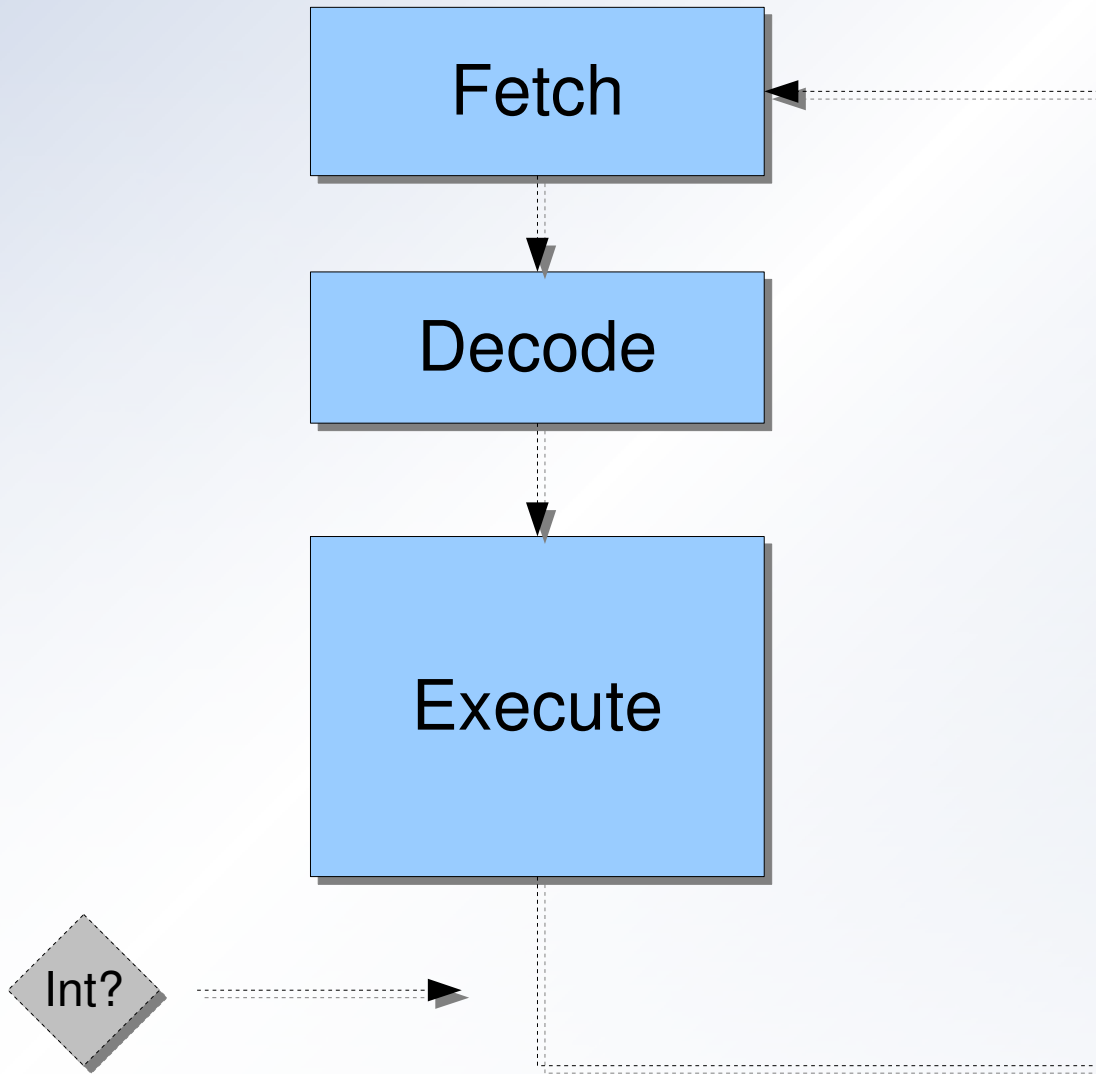
Lecture Outline

- Remind: Architettura di Von Neumann
- Sistema di Elaborazione
- Operazioni e Micro-operazioni
- Micro-operazioni
- Modello Generale di SE
- Microprogrammazione e Microprogrammi
- Microlinguaggi TS/PS

Architettura di Von Neumann



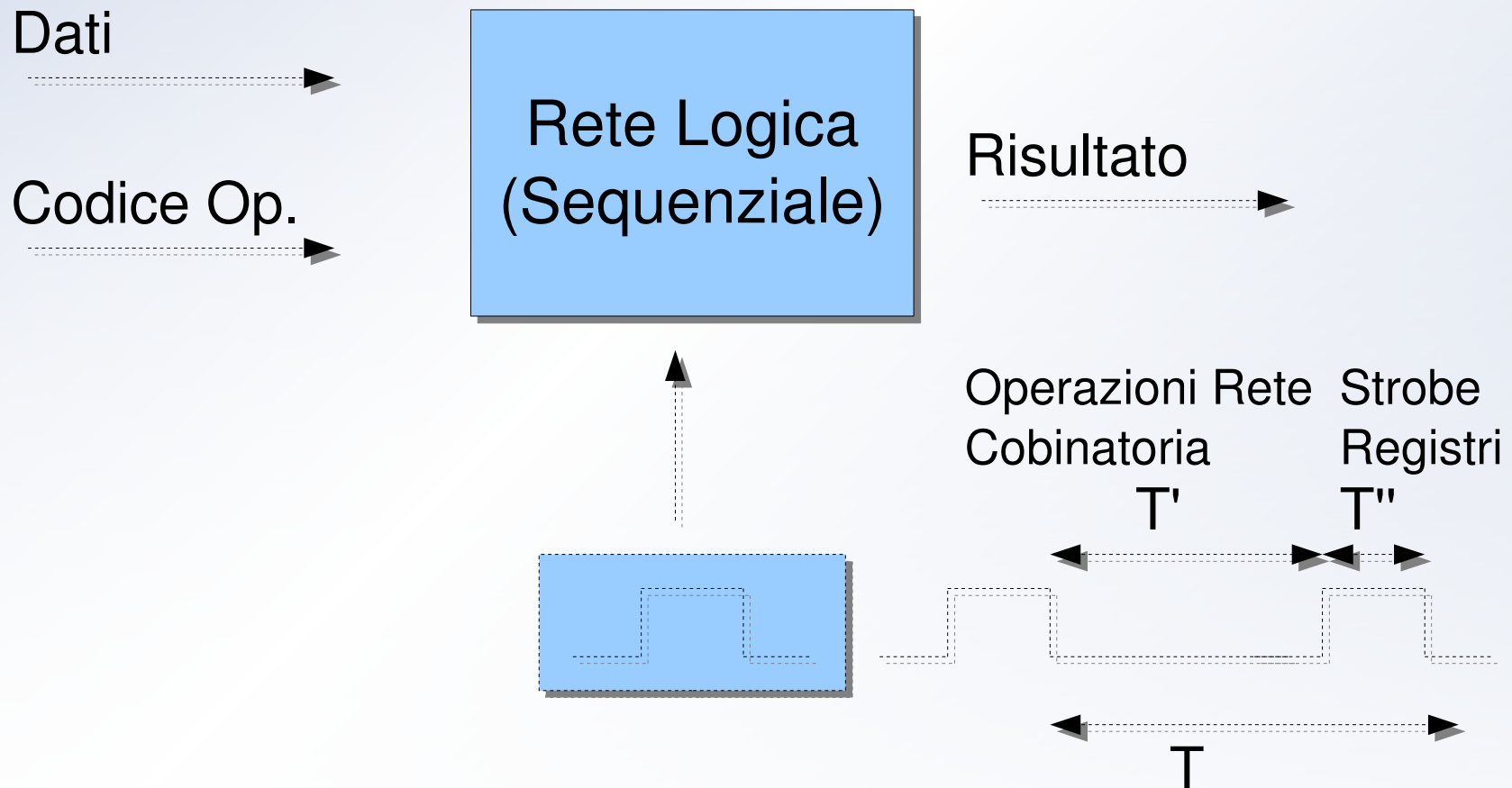
Architettura di Von Neumann



Sistema di Elaborazione

- Struttura **Logica** capace di eseguire **sequenze** di **operazioni**
 - Logica -> Rete Logica
 - Sequenze in generale qualsiasi
 - Nei sistemi Multiproc./Multicore vale ancora la definizione di Sequenza ?
 - Operazioni
 - **Operatore**
 - **Operando/i**
 - **Risultato**

Sistema di Elaborazione

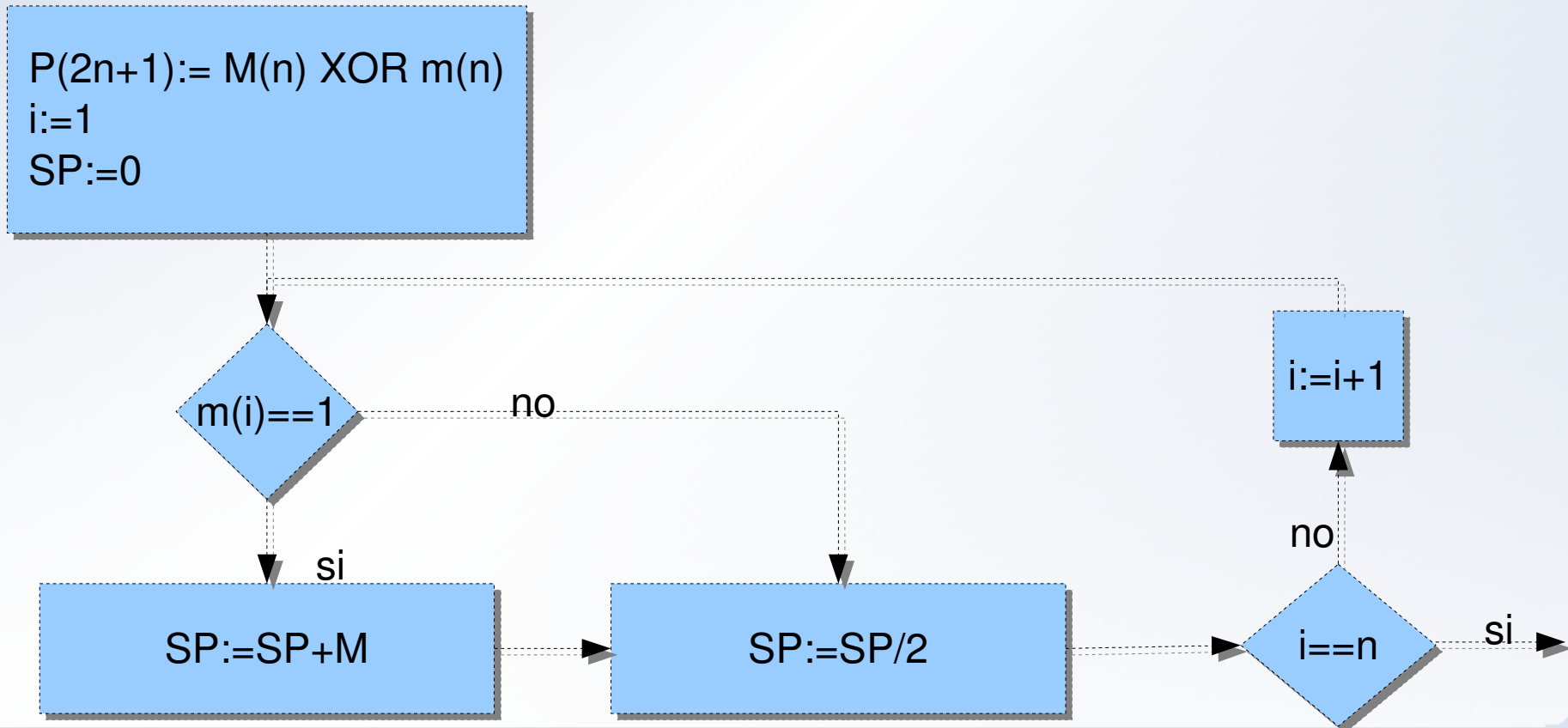


Operazioni e Micro-operazioni

- Operazione
 - Non Atomica
 - Eseguita in più **passi**
- **Micro-operazione**
 - Eseguita in un tempo elementare T
 - Trasferimento:
 - Consente di Sostituire il contenuto di un registro con l'informazione presente al suo ingresso (:=)
 - Una Micro-operazione può essere descritta da una serie di Relazioni di Trasferimento.
 - $M := A+M, B:= A$

Descrizione Operazioni di Sistema

- Linguaggi Opportuni
 - Diagrammi a Blocchi
- Es: Moltiplicazione 2 interi rappr. in segno e modulo

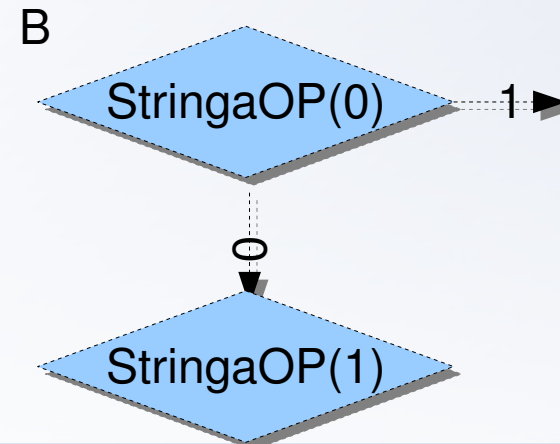
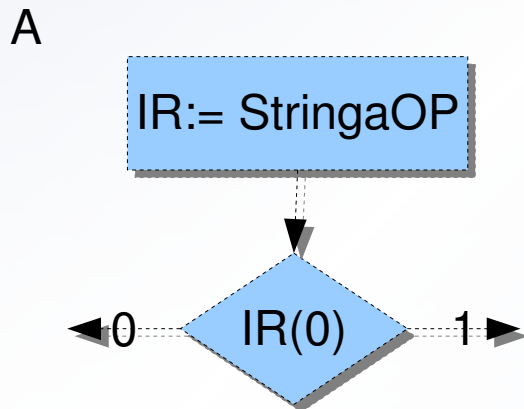


Operazione speciale e totale

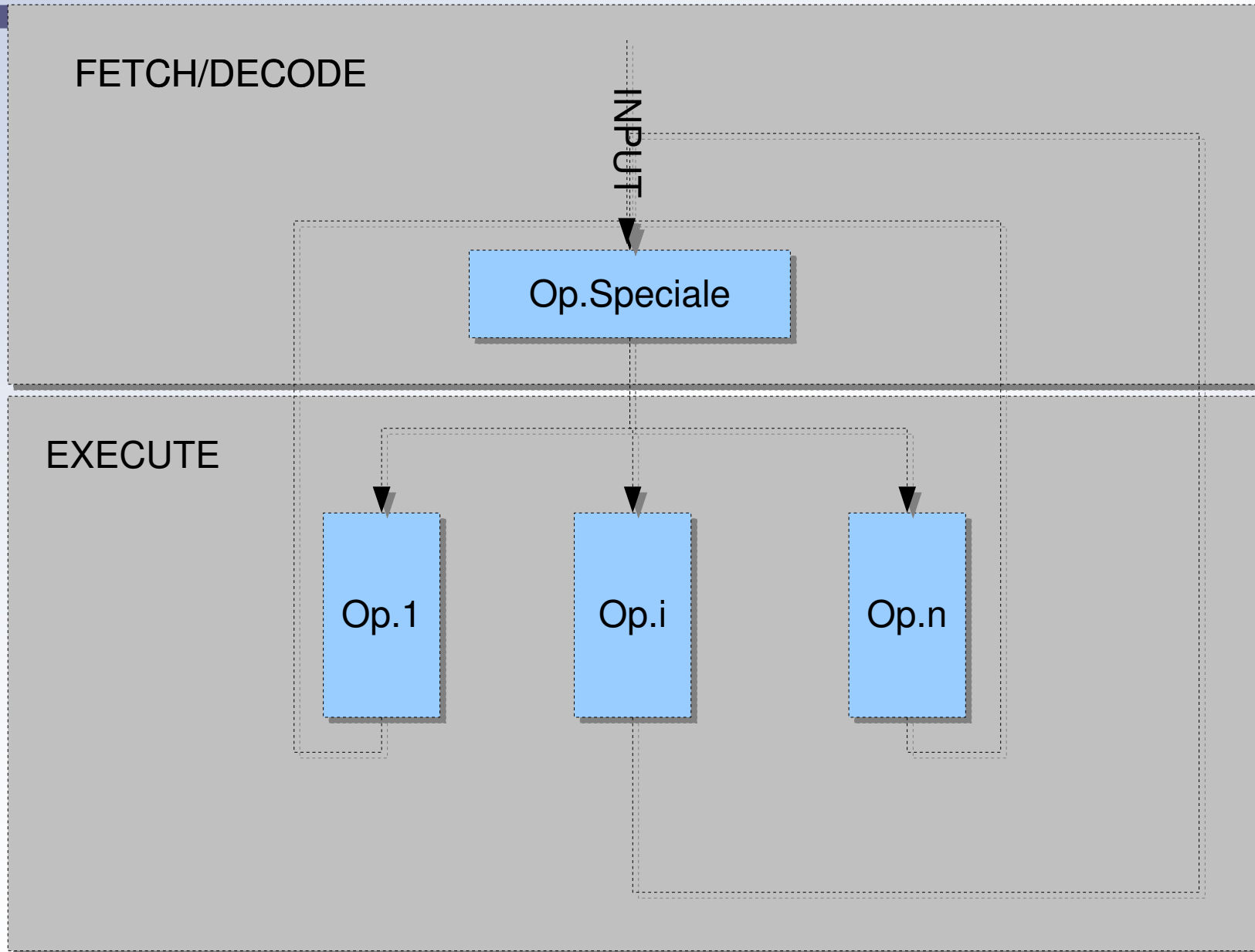
- Le operazioni in pratica sono **finite**
- Possono operare su un insieme **finito** di tipi di operandi
- Ogni Sistema di Elaborazione può essere definito anche tramite una **lista** delle possibili operazioni che può eseguire
- **Le Operazioni devono però essere sequenzializzate**
- Formalmente c'è bisogno di una operazione (non inclusa nella **lista**) che sequenzializzi le operazioni.
 - **OPERAZIONE SPECIALE**
 - Necessaria nei sistemi che hanno **ALMENO 2** Operazioni.
 - Esamina il Codice dell'operazione e ne ordina l'esecuzione sugli operandi in input

Tipi di Operazione Speciale

- Tipo A:
 - Memorizza l'operazione in un registro e quindi esegue la **decodifica**
- Tipo B:
 - Il codice dell'op. Deve essere presente nell'ingresso **ALMENO** per un tempo T.
 - Durante T si può avviare già la **decodifica** senza ricorrere ad ulteriori Registri



Operazione Totale



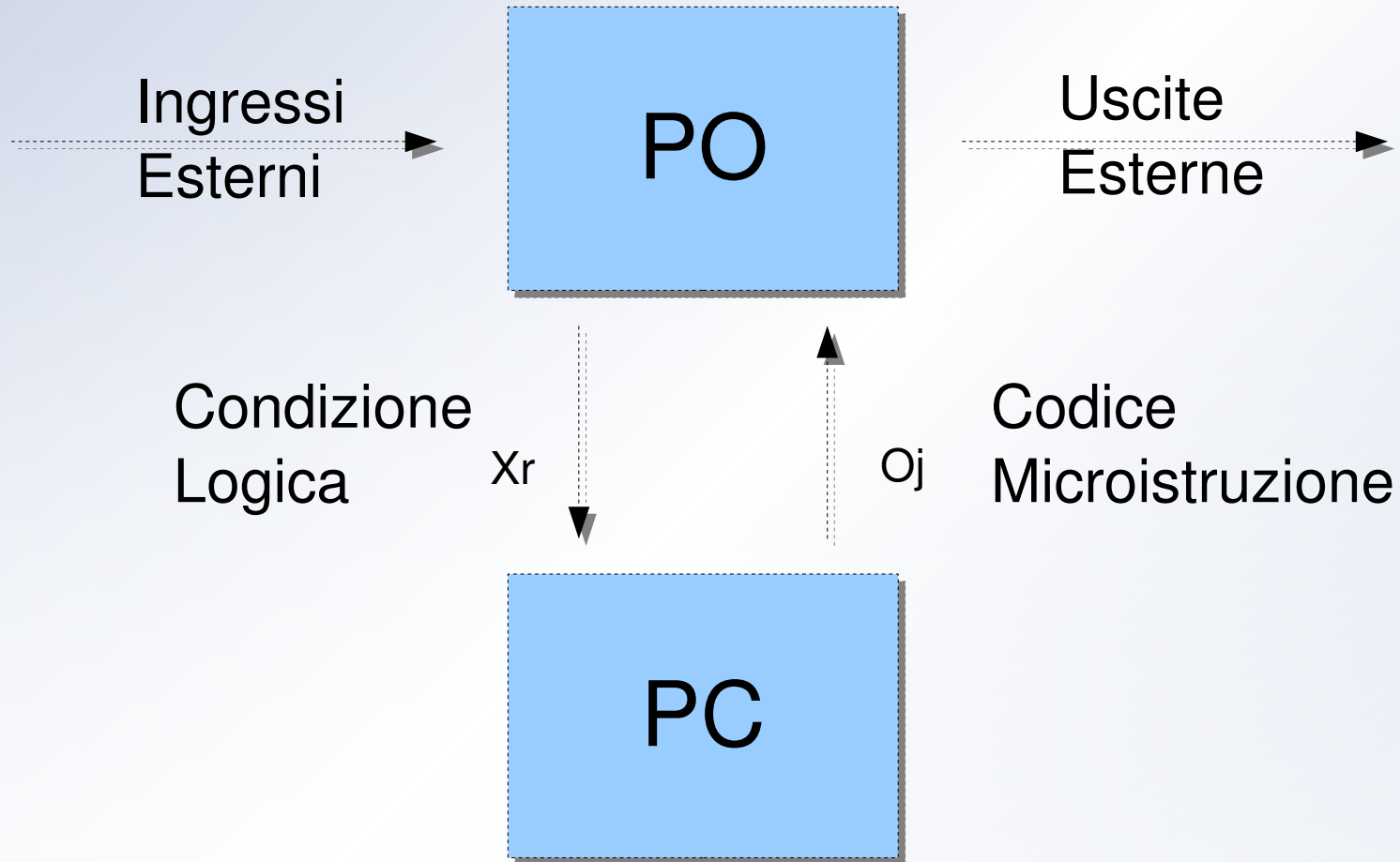
Operazione Totale: Caratteristiche Generali

- Il Diagramma ha **1 solo ingresso e nessuna uscita**
 - Il sistema funziona fino alla ... morte
 - Si suppone che il sistema sia sincronizzato con l'esterno in modo da avere sempre a disposizione il codice della prossima operazione da eseguire
- **Non devono esistere Cicli**
 - La decodifica e la parte di esecuzione devono essere organizzati ad **albero**.
- Se tra le operazioni esiste la **micro-operazione nulla** (lascia inalterati tutti i registri del sistema – e quindi il suo stato)
 - Possono esistere cicli con solo la micro-op nulla all'interno e almeno un blocco di decisione le cui condizioni siano funzione di informazioni esterne al sistema

Operazione Totale: Caratteristiche Generali

- Se l'op. Speciale è di **tipo A** e se i blocchi di esecuzione delle op. non contengono cicli, anche l'operazione totale non ha cicli.
- Se l'op. Speciale è di **tipo B**, possono crearsi dei cicli
 - Se l'uscita di un diagramma è connessa all'ingresso del diagramma dell'operazione speciale
 - In questo caso bisogna introdurre delle micro-op. nulle per eliminare i cicli nel blocco dell'operazione.

Modello Generale



Modello Generale

- Parte Operativa (PO)
- Parte di Controllo (PC)
- X_r
 - Codice della condizione logica che, assieme allo stato interno di PC, determina l'esecuzione della prossima micro-operazione
 - Determinato dalla configurazione delle variabili di condizione x_1, \dots, x_s .
- O_j
 - Codice della micro-operazione da eseguire.
 - Codifica i segnali di controllo per la PO per eseguire la micro-operazione.
 - Rappresentato dalla configurazione dei valori delle *variabili di controllo* a_1, \dots, a_m .
 - O0 (micro-operazione nulla)

Modello Generale

- X_r
 - A seconda del tipo di operazione Speciale
 - A: Vengono determinati a valle di un registro Istruzione
 - B: I Segali vengono inviati direttamente a PC.
- PO
 - E' sempre una rete sequenziale (contiene almeno un registro accumulatore)
 - Se i dati esterni e codice op. sono memorizzati nei registri e nessuna variabile di condizione dipende dalle variabili di controllo,
 - PO è di tipo Moore

Modello Generale

- PO e PC
 - Essendo sincronizzate sullo stesso clock formano un sistema a ciclo chiuso di reti sequenziali, quindi
 - Se PO è di tipo Moore, PC può essere di tipo Mealy o Moore
 - Se esiste almeno una variabile di condizione che dipende da variabili di controllo (PO è Mealy), PC può essere solo di tipo Moore
 - PERCHE'?

Modello Generale

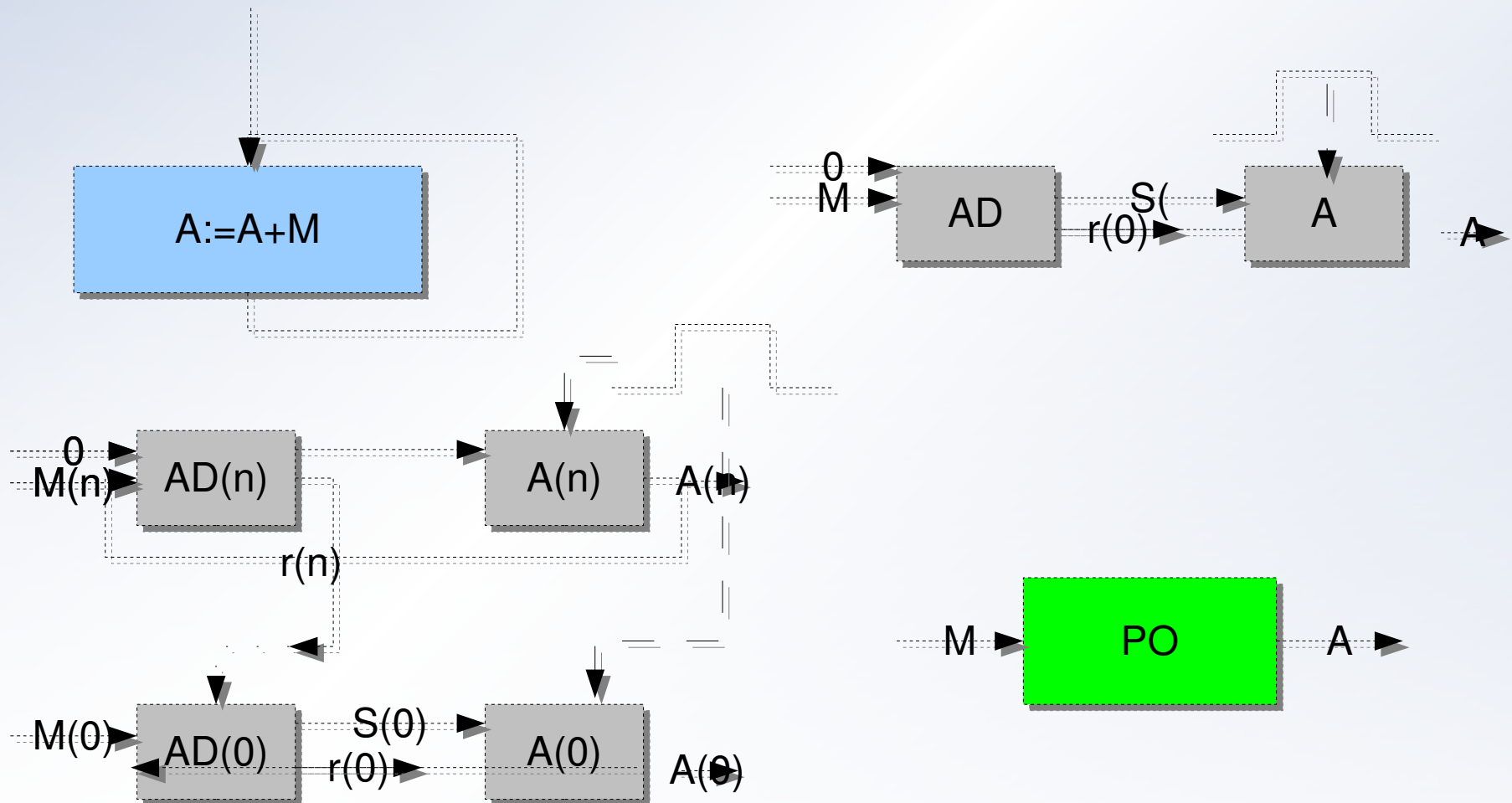
- PO e PC
 - Essendo sincronizzate sullo stesso clock formano un sistema a ciclo chiuso di reti sequenziali, quindi
 - Se PO è di tipo Moore, PC può essere di tipo Mealy o Moore
 - Se esiste almeno una variabile di condizione che dipende da variabili di controllo (PO è Mealy), PC può essere solo di tipo Moore
 - **In caso contrario, si avrebbe una cascata di 2 Mealy a ciclo chiuso... Nascono problemi di sincronizzazione nella propagazione dei segnali di ingresso/uscita.**

Tipologia PO/PC

- Moore-Mealy
- Moore-Moore
- Mealy-Moore
- Mealy-Mealy
 - In Pratica non esistono
 - Dovrebbero tutti utilizzare l'op. Speciale di tipo B.
 - Possono essere assimilate a reti Moore-Mealy per quanto riguarda il loro funzionamento.

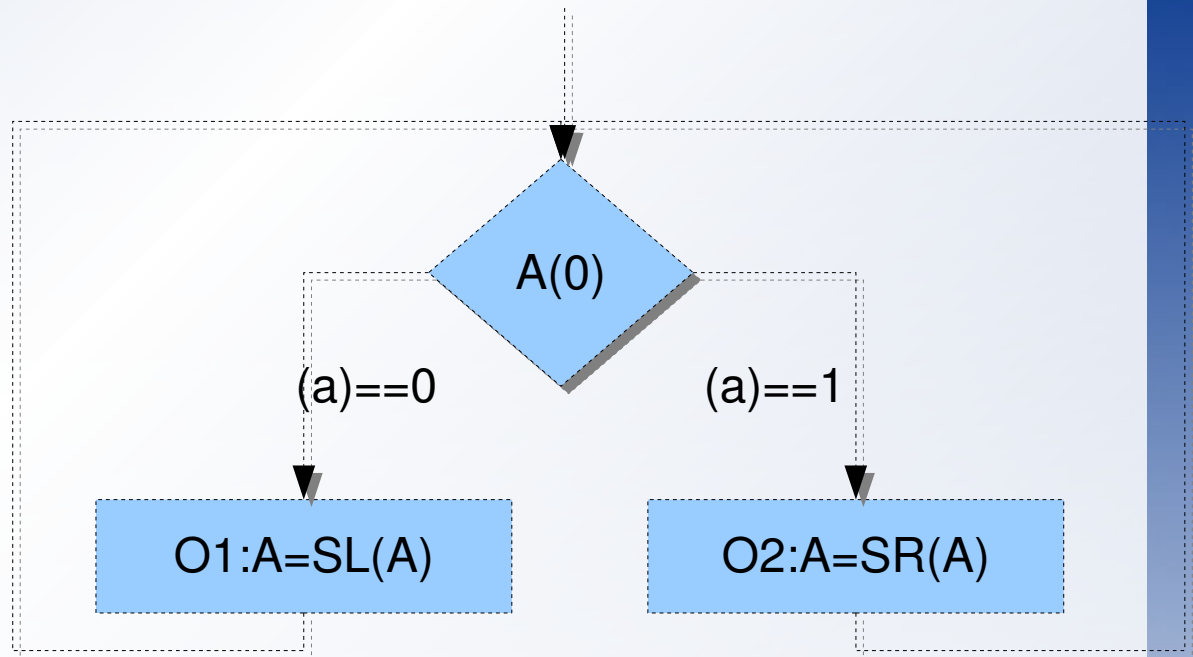
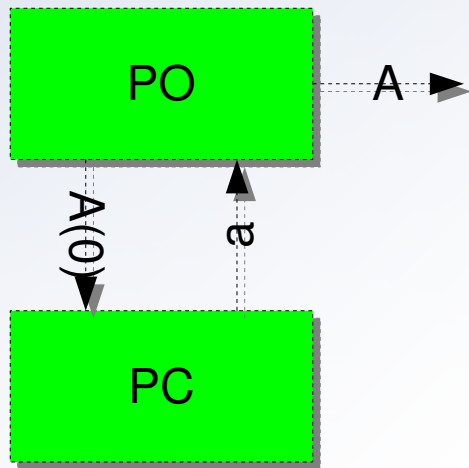
Sistemi Semplici

- Eseguono ogni micro-operazione in un tempo elementare



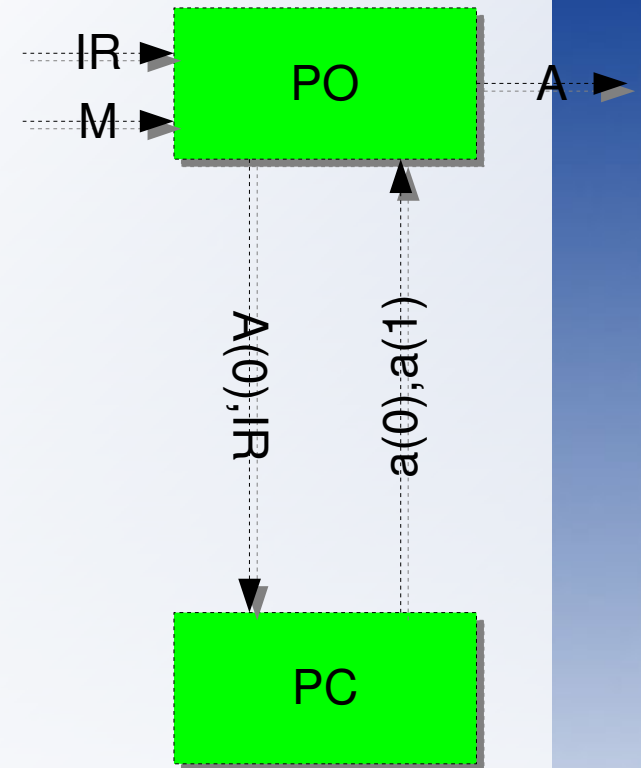
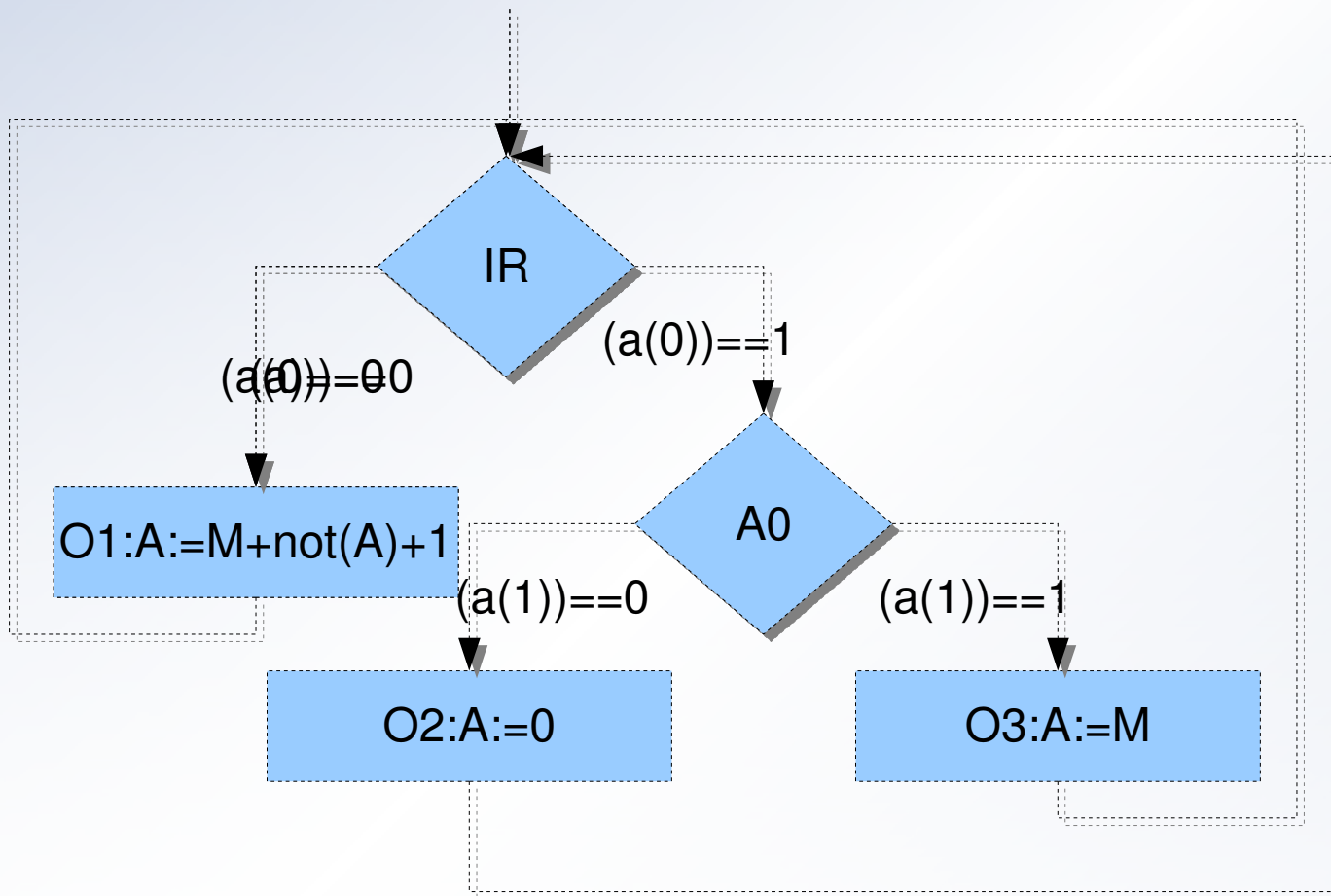
Sistemi Semplici

- Shift a Destra o Sinistra (PO/PC)



Sistemi Semplici

- Op1: $A := M + \text{not}(A) + 1$ (e' elementare ???)
- Op2: $\text{if } (A \% 2 == 0) A := 0 \text{ else } A := M$



Micro-programmazione e Micro-linguaggi

- Il Diagramma dell'operazione totale non descrive bene il comportamento del sistema
- **Micro-istruzione:** descrizione delle funzioni che PO-PC devono eseguire in un Passo Elementare
- **Micro-programma totale:** la *sequenza* di microistruzioni che descrivono l'operazione totale.
- Il Diagramma può essere visto come uno degli elementi del progetto di un Micro-Programma
- La Micro-Programmazione favorisce:
 - Riuso, testing
 - Scrittura di micro-programmi
 - ...

Componenti Micro-linguaggi

- **Micro-ordine**
 - O_j : Esegui O_j ($j=0..p$)
- **Trasferimento**
 - mk : Vai alla generica istruzione mk (microistruzione successiva)
- **Espressione Condizionale**
 - Cr : if (Cr) then ... (Cr : Condizione logica $Cr=f(x_1, \dots, x_s)$)
- **Frase:**
 - $\langle\langle O_j, mk \rangle\rangle$
- **Frase Condizionata**
 - $\langle\langle (Cr) O_j, mk \rangle\rangle$
- **Trasferimento Condizionato**
 - $\langle\langle (Cr) , mk \rangle\rangle$

Micro-linguaggi PS/TS

- Micro-linguaggi **ps** (Phrase Structured)

- Contengono **frasi condizionate**

- $\mu_h : (C_1)O_j^1, \mu_k^1 ; (C_2)O_j^2, \mu_k^2 ; \dots ; (C_q)O_j^q, \mu_k^q$

- Micro-linguaggi **ts** (Transfer Structured)

- Contengono **trasferimenti condizionati**

- Analoghi ai linguaggi macchina tradizionali

- $\mu_h : O_j^1, (C_1)\mu_k^1 ; O_j^2, (C_2)\mu_k^2 ; \dots ; O_j^q, (C_q)\mu_k^q$

- Micro-Linguaggio **M** :

- E' TS:

- Contiene i tre tipi di microistruzioni:

- $\mu_k \mid O_v, \mu_{k+1}$

- $\mu_k \mid O_0, \mu_h$

- $\mu_k \mid O_0, (C_r) \mu_{h+1} ; (\text{not } (C_r)) \mu_k$

Microprogrammi e Automi

- Per implementare il microprogramma, questo deve avere una corrispondenza con un automa a stati finiti
- X_r : Stati di Ingresso, O_j , Stati di uscita e interni

	X1	X2		Xq
mh	$m(k,1), O(j,1)$	$m(k,2), O(j,2)$		$m(k,q), O(j,q)$

PS -> PC tipo Mealy

	X1	X2		Xq
mh	$m(k,1)$	$m(k,2)$		$m(k,q)$

TS -> PC tipo Moore

Microprogrammi ed Automi

- Si suppone che le variabili di condizione e le condizioni logiche che le impiegano siano funzione solo di contenuti di registri e/o di informazioni applicate agli ingressi del sistema, ma non delle variabili **aj** che controllano l'esecuzione delle microoperazioni (chiamiamo quelle che dipendono dalle ai Xa e XR le altre)
- Se tutte le variabili di condizione sono XR, PO è di tipo Moore
 - E' possibile **anticipare** la disponibilità dei risultati trasformando PO in una macchina di tipo Mealy
 - Questa trasformazione produce una trasformazione XR->Xa

Esempi di Microprogrammi

- m1 | O0 (IR(0)==1) m(1+1); (IR(0)==0) m5
- m2 | O0 (SA==1) m(2+1); (SA==0) m1
- m3 | O4, m(3+1)
- m4 | O0, m1
- m5 | O1, m(5+1)
- m6 | O0, (C(0))==1) m(6+1); (C(0) ==0) m1
- m7 | O0, (Q(n)==1) m(7+1); (Q(n)) ==0) m9
- m8 | O2, m(8+1)
- m9 | O3, m(9+1)
- m10 | O0, m6
- Esercizio: Ricavare il diagramma a blocchi
 - A Cosa servono le O0????

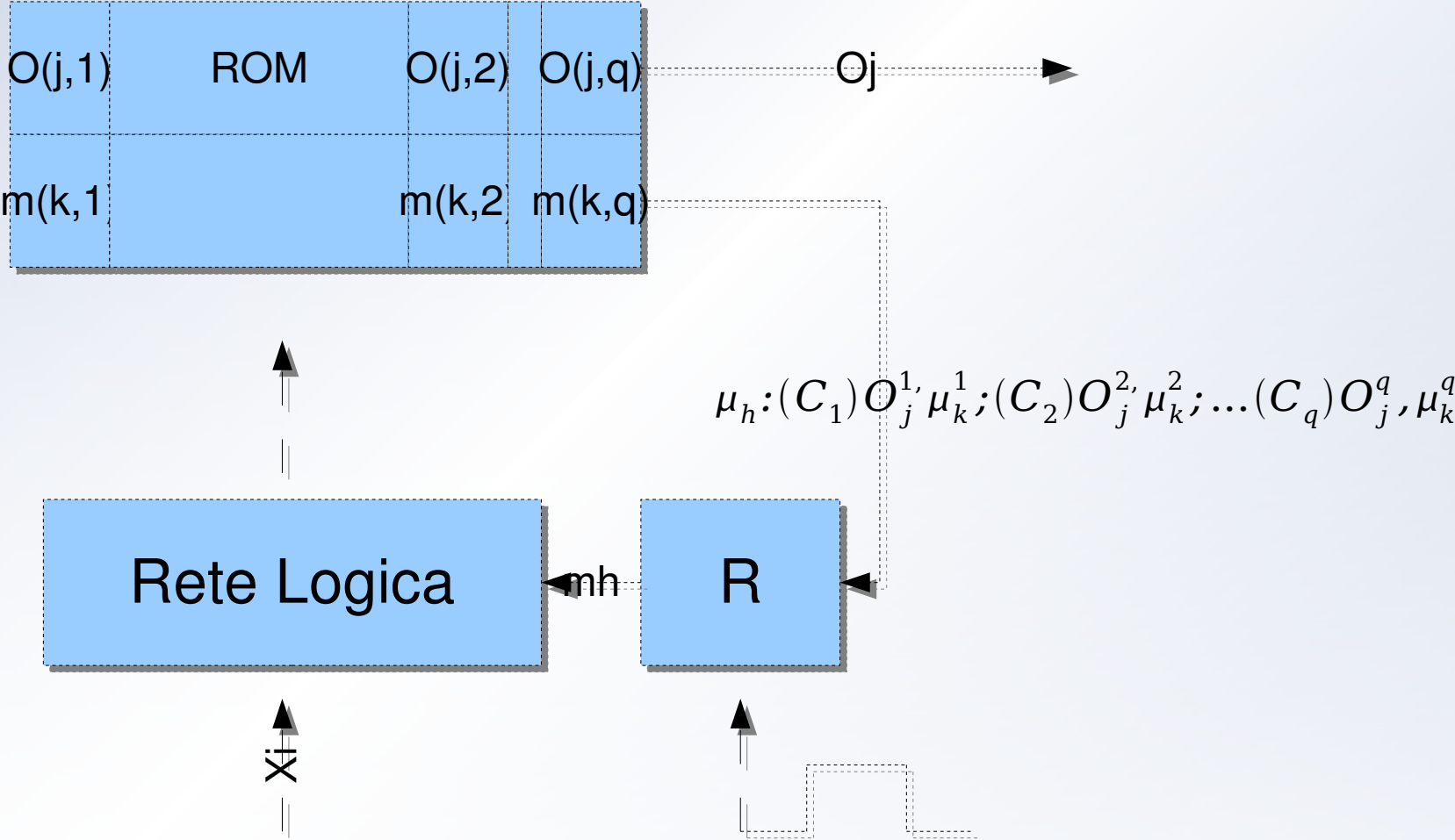
Realizzazione di Sistemi di Elaborazione

- PO
 - Si tratta di progettare un sistema integrato con le tecniche apprese durante il corso di Reti Logiche
- PC
 - Controllo Cablato
 - Anche questo progettato con le tecniche apprese durante il corso di Reti Logiche
 - Realizzazione di un rete sequenziale
 - Praticamente impraticabile quando il numero di micro-istruzioni/condizioni aumenta
 - Controllo Microprogrammato
 - Rete Logica +
 - Memoria di Controllo in cui è codificato il micro-programma

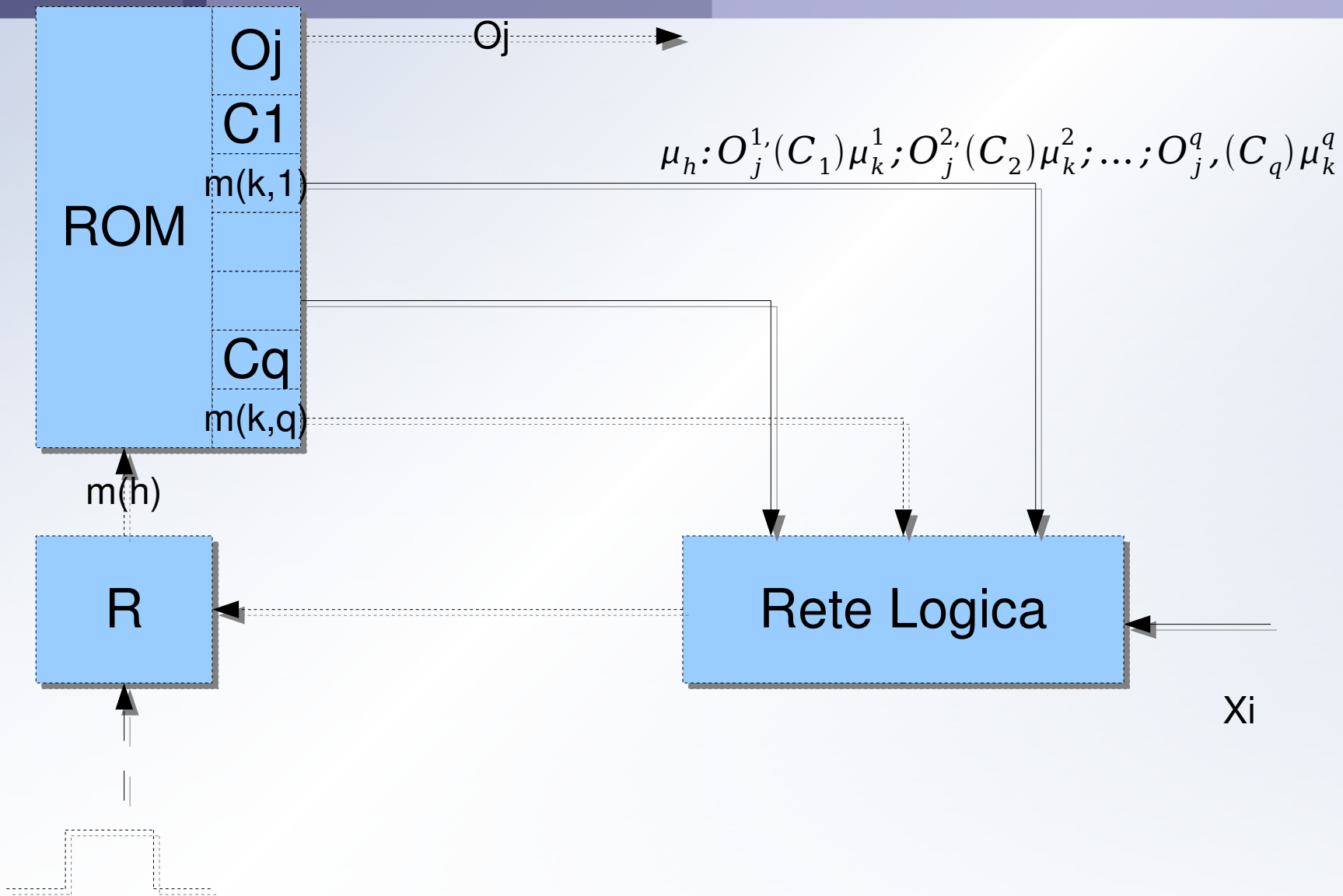
Controllo Microprogrammato

- **Memorizzazione di frase (mf):**
 - Controllo Micro-programmato per linguaggi **ps**.
- **Memorizzazione di micro-istruzione (mm):**
 - Controllo micro-programmato per linguaggi **ts**

Controllo mf



Controllo mm





33

Your Name
Your Title

Your Organization (Line #1)
Your Organization (Line #2)