

Reti sequenziali algoritmiche

Verso il microprocessore

(8.1-8.2)

Parte operativa e parte di controllo

Descrizione di reti algoritmiche

Sequenziatori con ROM e registro

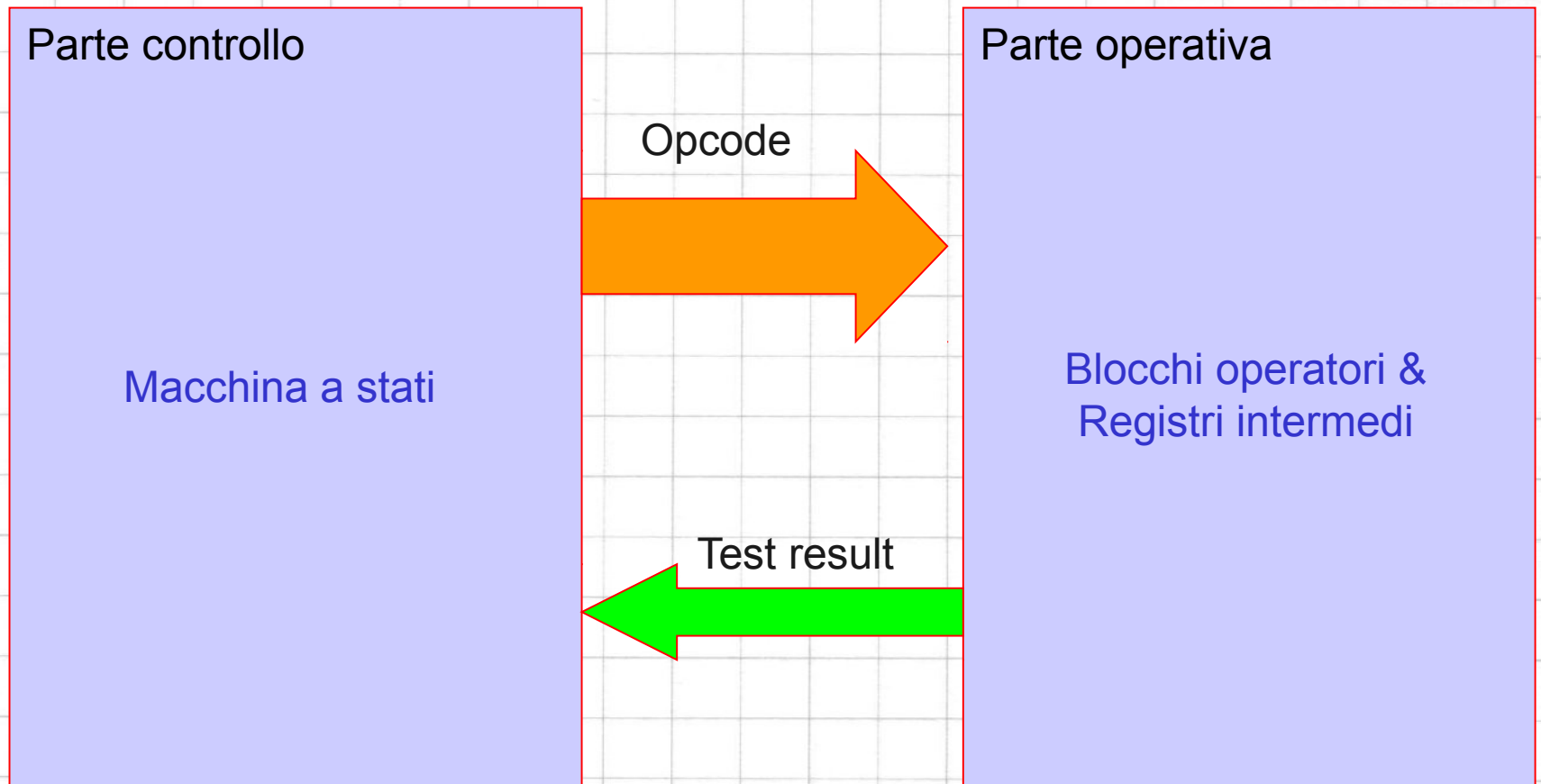
Sequenziatori con ROM e contatore

Evoluzione delle reti sincrone

Separazione tra parte **operativa** e di **controllo**

- La parte operativa elabora le informazioni
 - Operazioni logiche, aritmetiche
 - Test, confronti matematici
 - Appoggia i risultati intermedi in registri con abilitazione
- La parte di controllo gestisce il flusso di esecuzione delle operazioni in sequenze assegnate
 - Decide la sequenza degli stati percorsa dalla macchina (stato futuro)
 - Assegna i compiti alla parte operativa (codice operativo)
 - La sequenza dipende dai risultati di test e confronti (flag)

Parte operativa e di controllo



Un esempio

- Parte operativa
 - Composta da ALU e diversi registri, controllata dalla parola $Op(i)$
 - Sono disponibili i risultati di due test, F e G
- Parte controllo
 - Consideriamo una macchina a 8 stati
 - Consideriamo una sequenza costituita da sole alternative binarie
 - Lo stato futuro potrà essere uno tra due soli stati possibili
 - Il criterio di scelta per ogni passo sarà costituito dal valore di F oppure di G
 - Se i test danno esito positivo, la sequenza è ciclica e completa

Il microcodice

- Una descrizione alternativa
 - Lo stato è rappresentato da una label
 - Il codice operativo da una istruzione di assegnazione
 - Il test e il corrispondente stato futuro da una istruzione del tipo
 - if (test) then (stato futuro se vero) else (stato futuro se falso)

```
S0: opcode = Op0; if F then S1 else S3
```

Nuova descrizione

```
S0: opcode = Op0; if F then S1 else S3
S1: opcode = Op1; if F then S2 else S0
S2: opcode = Op2; if G then S3 else S2
S3: opcode = Op3; if F then S4 else S4
S4: opcode = Op4; if F then S5 else S6
S5: opcode = Op5; if G then S6 else S7
S6: opcode = Op6; if F then S7 else S2
S7: opcode = Op7; if G then S0 else S1
```

Architettura della parte di controllo

- Queste macchine sono dette **sequenziatori**
- Possono essere sintetizzate come macchine di Moore o Mealy ritardate
 - Gli ingressi sono i valori F e G
 - Le uscite sono il codice operativo
- L'architettura deriva dal microcodice
 - Si codificano gli stati con numeri binari in sequenza
 - Si inseriscono le istruzioni del microcodice in una ROM
 - Si usa un registro per l'indirizzo
 - Si usano multiplexer per la scelta dei test e dello stato futuro

Codifica degli stati e dei flag

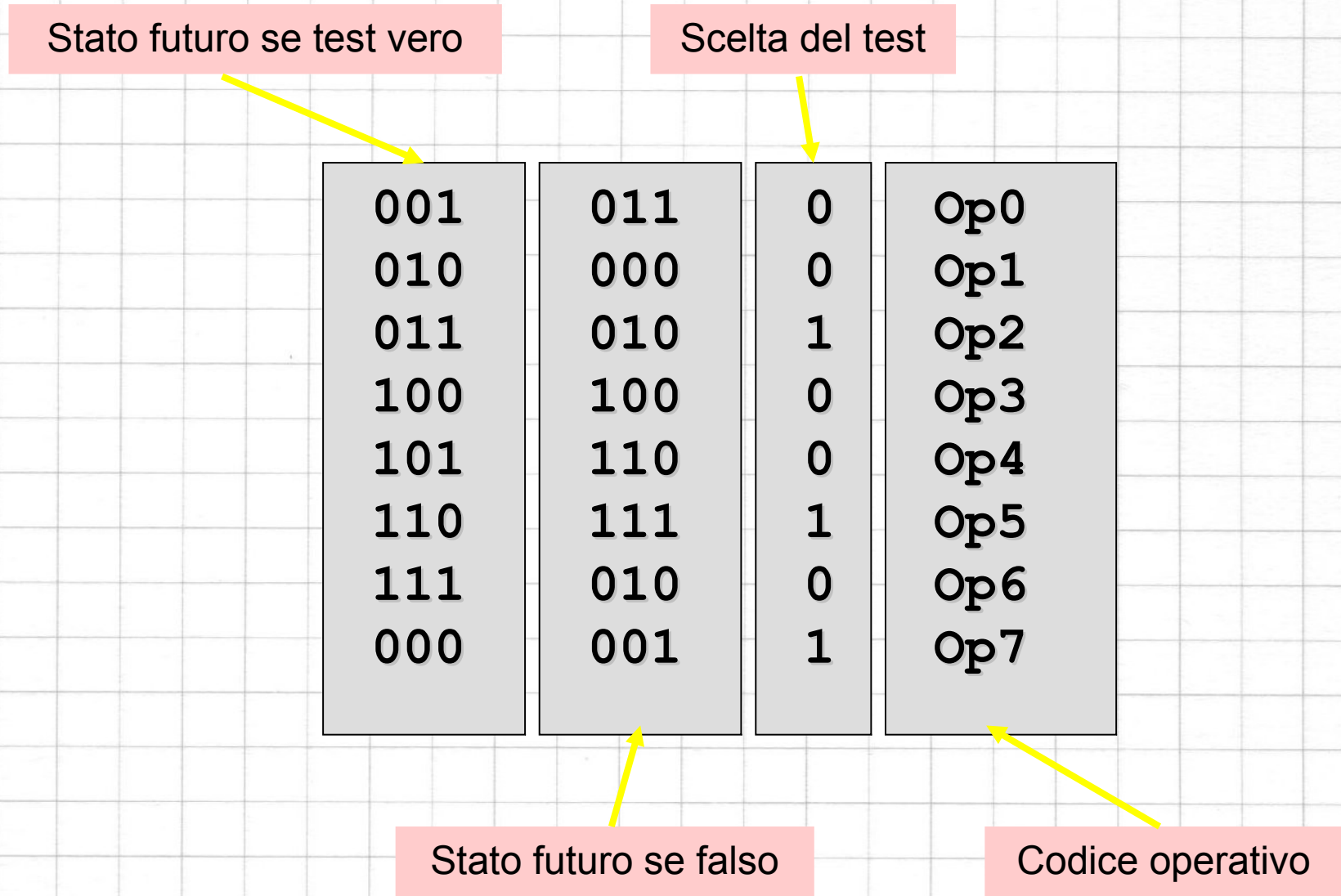
Stati

S0 = 000
S1 = 001
S2 = 010
S3 = 011
S4 = 100
S5 = 101
S6 = 110
S7 = 111

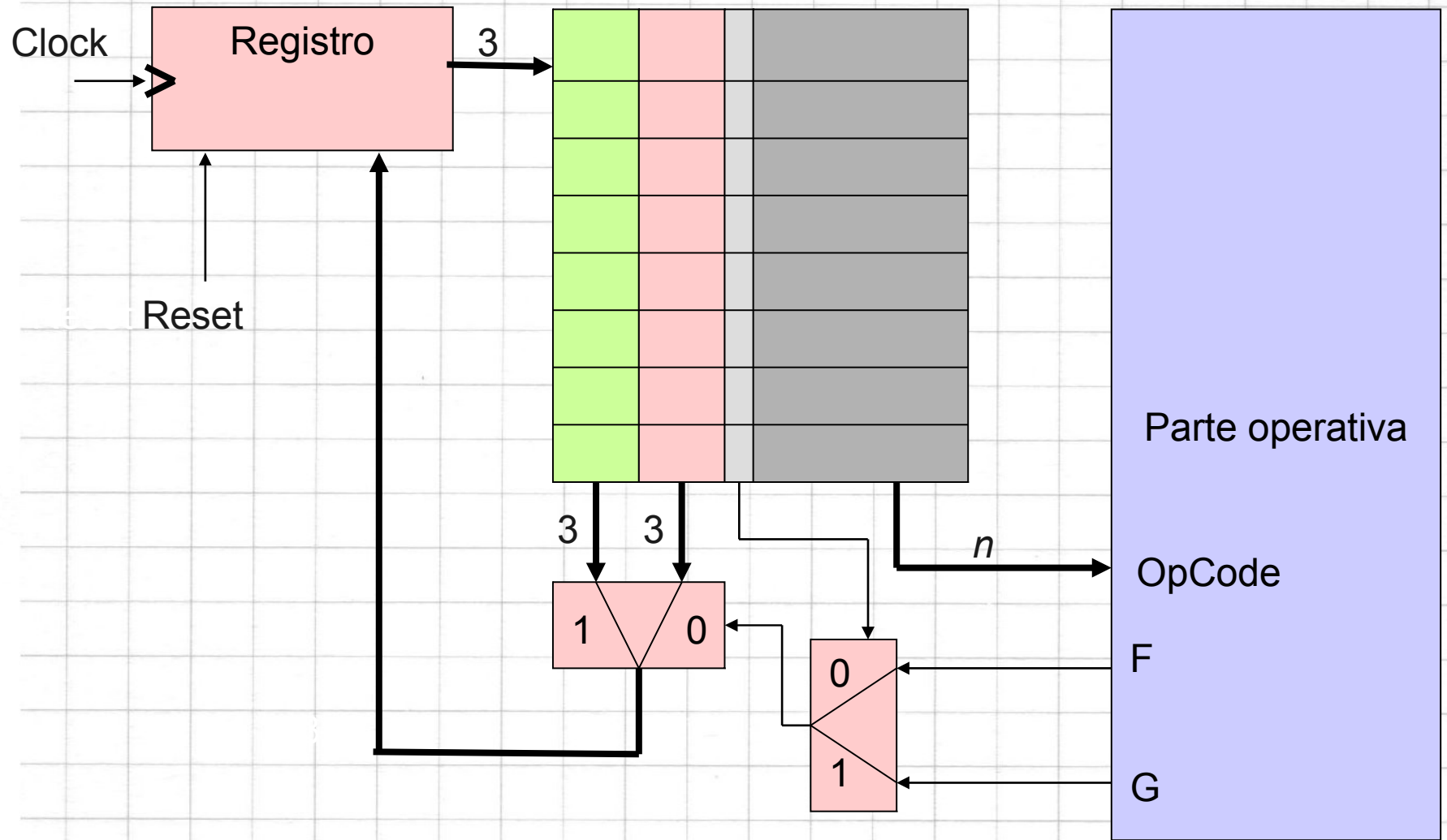
Flag

F = 0
G = 1

Contenuto della ROM (8 righe)



Architettura a ROM



Una possibile semplificazione

- Se con test positivo la sequenza seguita è ciclica e completa, la ROM si può semplificare
 - Si codificano gli stati con codici consecutivi, pari agli indirizzi delle righe della ROM
 - Cioè 2^N , dove N è il numero delle variabili di stato
 - Si sostituisce il registro con un contatore con caricamento parallelo
 - Si invia il flag selezionato all'abilitazione del contatore al caricamento
 - 1: Count (C)
 - 0: Load (\bar{L})

Architettura con contatore

