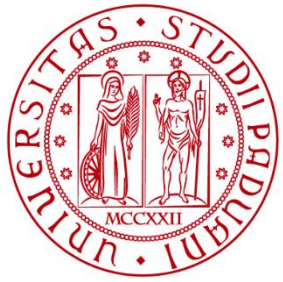
DEI
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sistemi Digitali

Tecnologie di implementazione dei circuiti digitali

Marta Bagatin, marta.bagatin@unipd.it

Corso di Laurea in Ingegneria dell'Informazione

Anno accademico 2022-2023

Scopo della lezione

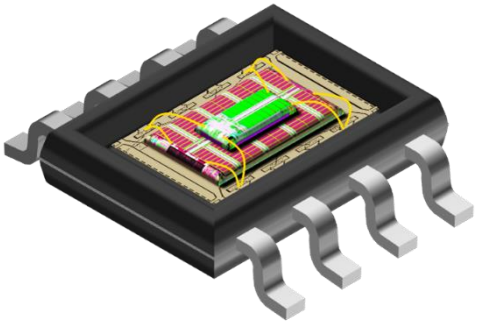
- Esaminare le possibili tecnologie (hardware) con cui può essere realizzato un circuito digitale e i principali parametri tecnologici
- Fare una panoramica delle diverse soluzioni disponibili per realizzare i dispositivi logici programmabili

Design space

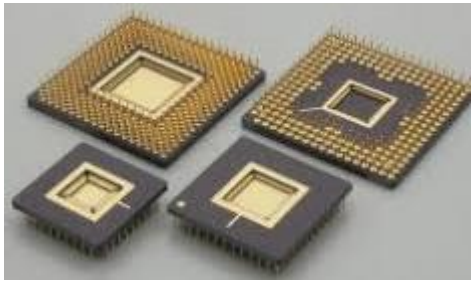
- Dato un sistema digitale, si definisce design space l'insieme delle tecnologie implementative che si sono scelte, cioè gli **elementi primitivi disponibili nella tecnologia scelta e i parametri che li caratterizzano**

Circuito integrato

- Un circuito integrato (Integrated Circuit, IC) è una porzione di silicio (chip) su cui sono costruite porte logiche ed elementi di memoria, interconnessi tra di loro all'interno del chip
  - Il chip è poi **montato su un package** e vengono realizzate apposite connessioni tra il chip e i pin del package (tra una decina e alcune centinaia)
  - Le caratteristiche di un IC sono specificate dal produttore in un documento detto **datasheet**



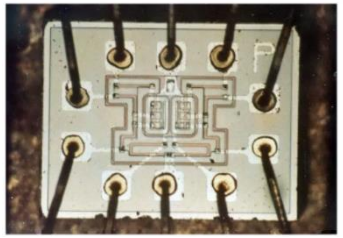
Source: www.embedded.com



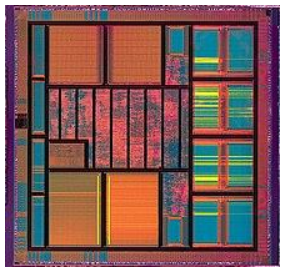
Source: www.ntktech.com

Livello di integrazione

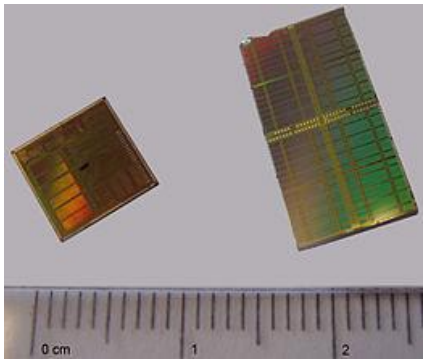
- Con l'evoluzione della tecnologia, il livello di integrazione degli IC è cresciuto, con un **numero sempre maggiore di porte logiche sullo stesso chip** (a parità di area)
- In base al numero di porte logiche, i circuiti si classificano in
  - **Small-Scale Integrated (SSI)**: < 10 porte logiche, ingressi e uscite collegati direttamente ai pin del package
  - **Medium-Scale Integrated (MSI)**: 10-100 porte logiche, realizzano specifiche funzioni logiche elementari (es. sommatore a 4 bit)
  - **Large-Scale Integrated (LSI)**: 100-10000 porte logiche (es. memorie o piccoli microprocessori)
  - **Very Large Scale Integrated (VLSI)**: > 10000 porte logiche (es. microprocessori complessi o chip per elaborazione di segnali digitali)



IC (NOR) usato a bordo del computer di controllo della navicella spaziale Apollo (~1961)



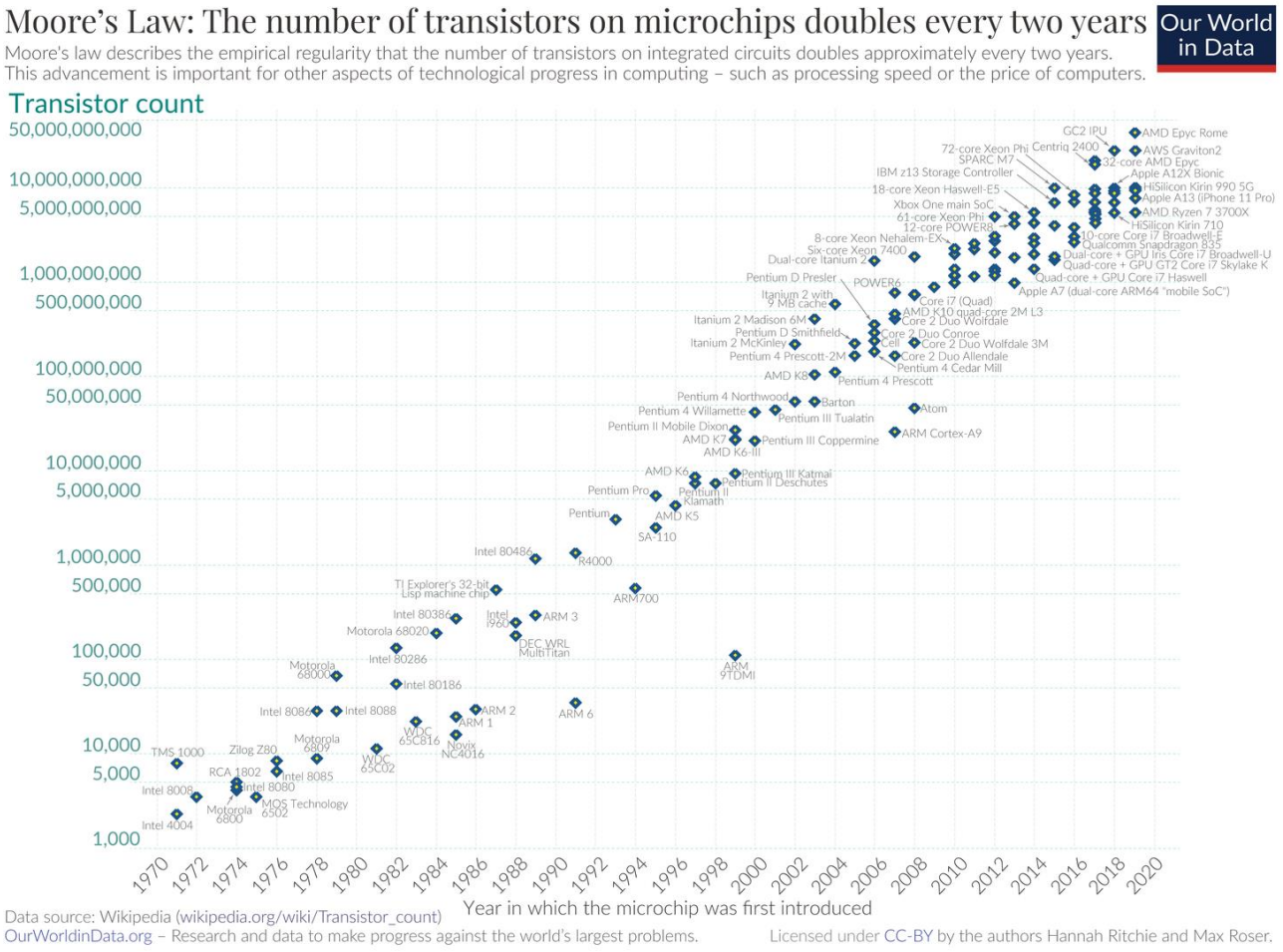
Atmel Diopsis 740 Dual-core DSP



Source: Wikipedia

Legge di Moore

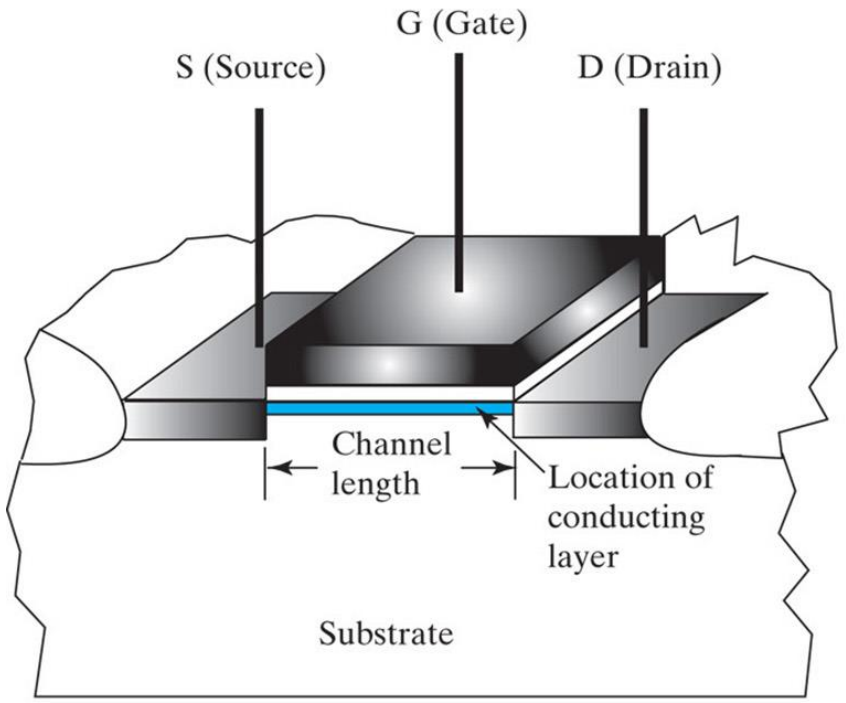
- 1965: Gordon Moore (Fairchild Semiconductor e co-fondatore di Intel) predisse che il **numero di componenti per circuito integrato sarebbe raddoppiato ogni anno**
- 1975: la previsione fu rivista (raddoppio ogni due anni)
- Da allora, questa previsione è stata utilizzata nell'industria dei semiconduttori per guidare la pianificazione e fissare gli obiettivi per la ricerca e lo sviluppo



Tecnologia CMOS

- Oggi la tecnologia più comunemente usata per l'implementazione dei circuiti integrati è la tecnologia CMOS (Complementary Metal Oxide Semiconductor)
- L'elemento alla base della tecnologia è il transistor MOS (Metal Oxide Semiconductor) realizzato con materiali come il silicio, con proprietà situate tra i conduttori e gli isolanti
- Argomenti saranno affrontati nei corsi del secondo anno (Elettronica, Elettronica dei Sistemi Digitali)

Dimensioni tipiche per la lunghezza di canale di un transistor moderno sono inferiori ai 10 nm (10^{-9} m)

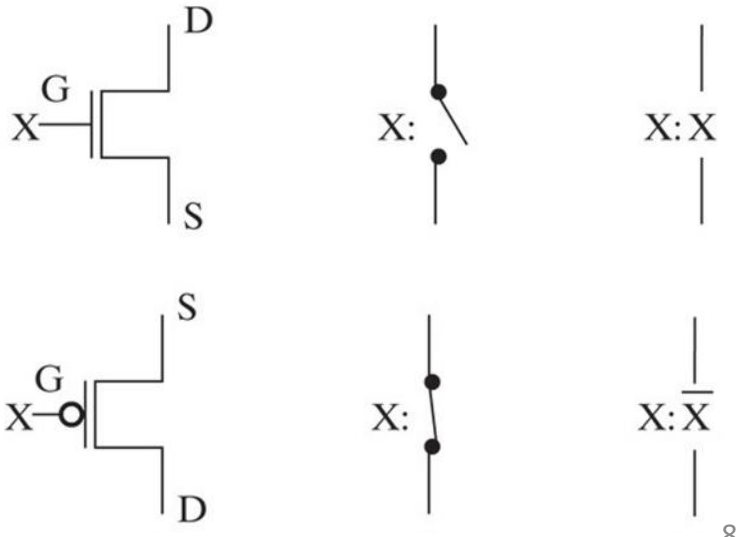


Transistor e modello a interruttore

- In un **sistema digitale** e in una **trattazione semplificata**, i transistor possono essere approssimati a interruttori: se aperti (OFF) non fanno passare un segnale, se chiusi (ON) lo fanno passare
- Questo modello del prim'ordine ignora dettagli fisici e complessità di questi dispositivi, catturandone solo il **comportamento logico**
- La tecnologia CMOS usa 2 tipi di transistor, che hanno comportamenti diversi perché realizzati con materiali diversi: la posizione dell'interruttore è controllata dal valore logico al terminale di Gate

  - MOS a canale n: **OFF** (interruttore aperto) se ha un **valore logico basso al Gate**, **ON** (interruttore chiuso) se ha un **valore logico alto al Gate**

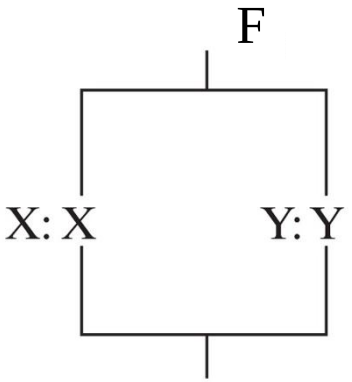
  - MOS a canale p: **OFF** (interruttore aperto) se ha un **valore logico alto al Gate**, **ON** (interruttore chiuso) se ha un **valore logico basso al Gate**



Logica a interruttore

- Un **circuito digitale** è composto da transistor che modellizziamo come interruttori: una funzione F viene realizzata facendo in modo che esista un percorso completo attraverso il circuito quando F deve essere uguale a ‘1’ e non esista quando F = ‘0’

  – Es. 1: circuito realizzato con transistor a canale n



F = ‘1’ se X = 1 **o** Y = 1

$$F = X + Y$$

Interruttori in serie realizzano porte AND, in parallelo realizzano porte OR

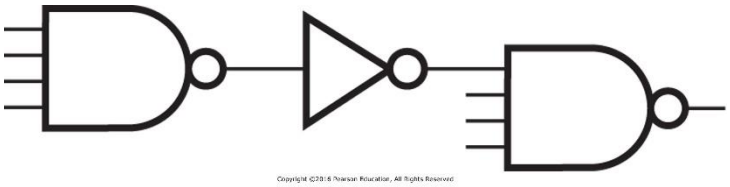
  – Es. 2: circuito realizzato con transistor a canale p

G = ‘1’ se e solo se cioè X = 0 **e** Y = 0

$$G = \overline{X} \cdot \overline{Y} = \overline{X + Y}$$

Parametri tecnologici

- **Fan-in**: numero di ingressi disponibili in una porta logica
  - Tipicamente il massimo fan-in è 5 per le porte logiche primitive in circuiti ad alta velocità: per realizzare porte con più di 5 ingressi, si usano porte multiple (con un numero inferiore di ingressi) connesse tra di loro
  - Es:



NAND a 7 ingressi realizzata con due NAND a 4 ingressi e un inverter

- **Fan-out**: quantifica quello che può essere collegato all'uscita di una porta logica senza degradare le prestazioni della porta stessa (detto anche carico pilotato da una porta logica)
  - Tipicamente si misura il fan-out in termini di numero di carichi standard (= inverter di dimensioni minime)
  - Es: un inverter minimo collegato all'uscita di una porta costituisce un carico standard unitario. Se una porta pilota 6 inverter minimi, ha un fan-out uguale a 6 carichi standard

Parametri tecnologici

- **Margine di rumore**: massimo rumore esterno che, sovrapposto al segnale di ingresso, non determina un cambiamento indesiderato dell'uscita del circuito
- **Ritardo di propagazione**: tempo che impiega il cambiamento del valore logico di un segnale a propagarsi dall'ingresso all'uscita di una porta logica. La velocità di un circuito è inversamente proporzionale al ritardo di propagazione della porta più lenta
- **Dissipazione di potenza**: è la potenza che viene consumata da una porta logica. Dato che la potenza è dissipata in forma di calore, la specifica sul massimo consumo di un circuito è funzione della temperatura operativa. Per sistemi alimentati a batteria, il consumo di potenza determina il tempo di vita della batteria

Parametri tecnologici: il costo

- Il **costo** di una porta logica è il contributo del costo della porta stessa rispetto al costo totale del circuito integrato che la contiene
- Il costo è direttamente proporzionale all'**area di silicio** occupata e dipende dal **numero e dalla dimensione dei transistor** e dalle connessioni all'interno della porta logica
- I costi per la progettazione di un circuito integrato si dividono in
  - Costi **Non Recurring Engineering (NRE)**: una tantum (design, verifica, test, …), vengono ammortizzati per grandi volumi di produzione
  - Costi di **produzione**: costi diretti, che devono essere sostenuti per ciascun chip prodotto (materiali, manodopera, energia, …)
  - Per volumi di produzione ridotti, i costi NRE dominano sui costi di produzione per ciascuna unità

Dispositivi logici programmabili (PLD)

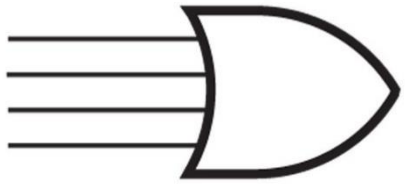
- Finora abbiamo parlato di implementazioni tecnologiche fisse, cioè fabbricate con uno o più circuiti integrati che implementano funzioni logiche predefinite e non modificabili
- Al contrario, i **dispositivi logici programmabili** (**PLD**) sono circuiti integrati fabbricati con strutture che possono essere **configurate dall'utente per svolgere diverse funzioni logiche**
  - La procedura di **configurazione «cambia» l'hardware**, determinando la particolare funzione logica che verrà implementata nel PLD
  - Da non confondere con la procedura di programmazione software in un microcontrollore!
- Vedremo 4 categorie di PLD:
  - Read-only Memory (**ROM**)
  - Programmable Logic Array (**PLA**)
  - Programmable Array Logic (**PAL**)
  - Field Programmable Gate Array (**FPGA**)

Tecnologie per la logica programmabile

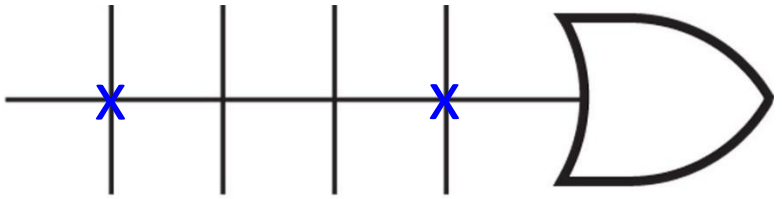
1) Tecnologie che comportano un **cambiamento irreversibile** nella loro struttura al momento della configurazione, possono essere configurate una sola volta
   - Fuses/anti-fuses: i collegamenti tra gli elementi logici vengono eliminati/ creati in modo selettivo applicando delle tensioni elettriche elevate
   - Mask programming: i collegamenti vengono realizzati dal costruttore durante l'ultima fase del processo di fabbricazione del chip
2) Tecnologie la cui configurazione determina un **cambiamento reversibile** e che possono essere riconfigurate molteplici volte
   - Basate su interruttori connessi ad elementi di memoria volatile (SRAM). Le informazioni sulla programmazione sono perse quando l'alimentazione viene rimossa
   - Basate sul controllo dell'accensione dei transistor, che hanno soglie 'configurabili'. Tipicamente non-volatili, mantengono la programmazione anche in assenza di alimentazione

Simbologia

- Le porte logiche in un PLD possono avere un numero elevato di ingressi
  - Per indicare porte logiche con fan-in elevati nei dispositivi a logica programmabile useremo una simbologia compatta:



(a) Conventional symbol



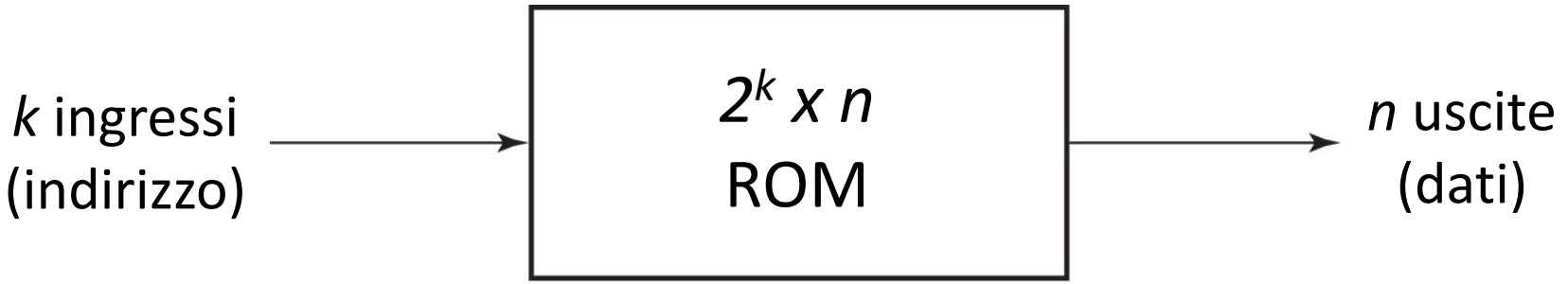
(b) Array logic symbol

  - La presenza di una x all'intersezione delle linee indica che c'è connessione, l'assenza della x indica che quella linea non è connessa

Read Only Memory (ROM)

- In una ROM (**memoria di sola lettura**) le informazioni sono memorizzate in **modo permanente**, anche in assenza di alimentazione (**memorie non volatili**). Le ROM programmabili (PROM) possono essere cancellate e riprogrammate tramite processi particolari, che vengono effettuati più raramente dei normali processi di lettura e scrittura nelle restanti classi di memorie (RAM: Random Access Memory, memorie a lettura e scrittura)
- Una ROM può essere realizzata con diverse tecnologie
  - **ROM**: mask programmed (programmata nella foundry)
  - **PROM**: fuse/anti-fuse (programmata dall'utente una sola volta)
  - **EPROM** (Erasable and Programmable): cancellabile tramite raggi ultravioletti, può essere cancellata e riprogrammata dall'utente
  - **EEPROM o E^2PROM** (Electrically Erasable and Programmable): cancellabile elettricamente dall'utente e riprogrammabile
  - **Memoria Flash**: evoluzione della E^2PROM
- La scelta della tecnologia dipende da volumi di produzione, modalità di utilizzo (es: frequenza di riprogrammazione) e prestazioni (es: velocità)

ROM: schema a blocchi



- L'**ingresso** contiene l'**indirizzo** del dato (word)
- L'**uscita** contiene il **dato** selezionato dall'indirizzo fornito in ingresso
- Una ROM può memorizzare $\boldsymbol{2^k}$ **word da** $\boldsymbol{n}$ **bit**
- Essendo una **memoria di sola lettura**, non c'è la possibilità di scriverla, cioè di fornire dati in ingresso

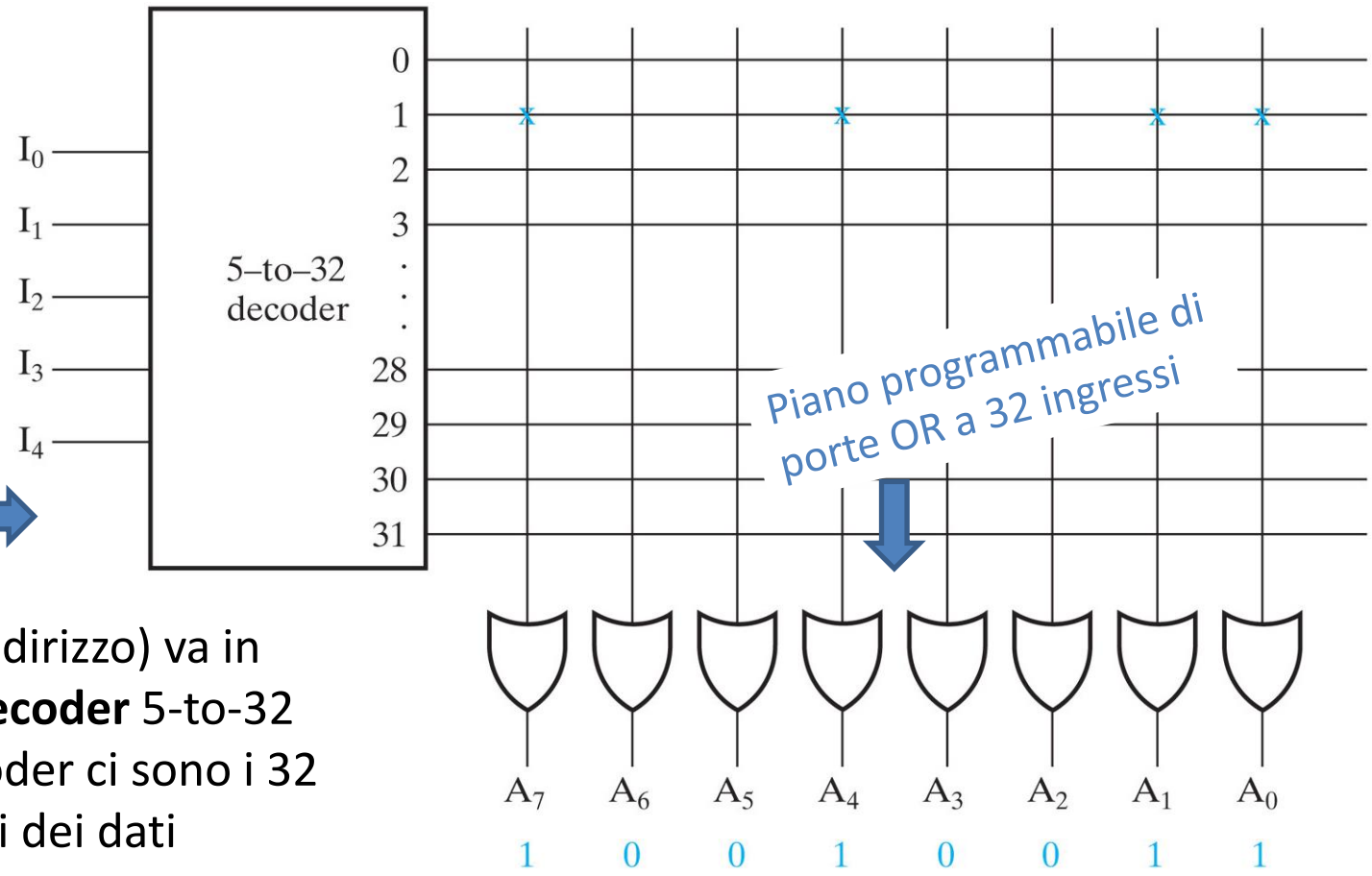
Idea concettuale di memoria

Memory Address		Memory Contents
Binary	Decimal	
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
	.	.
	.	.
	.	.
	.	.
	.	.
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

FIGURE 7-2
Contents of a 1024 × 16 Memory

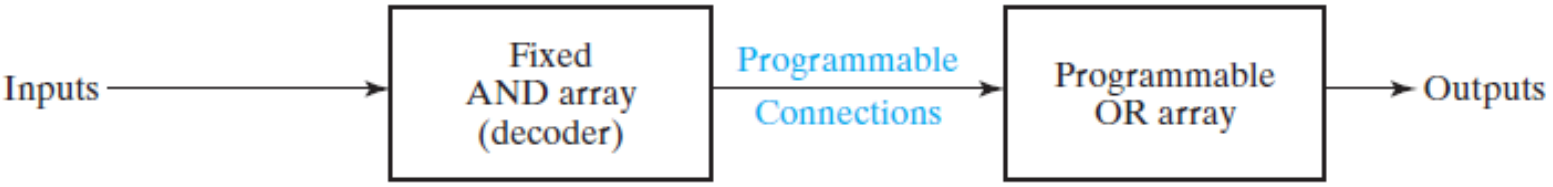
ROM: Esempio (32 parole da 8 bit)

Una **ROM 2^k x n** è costituita da un decoder k-to-2^k e da n porte OR

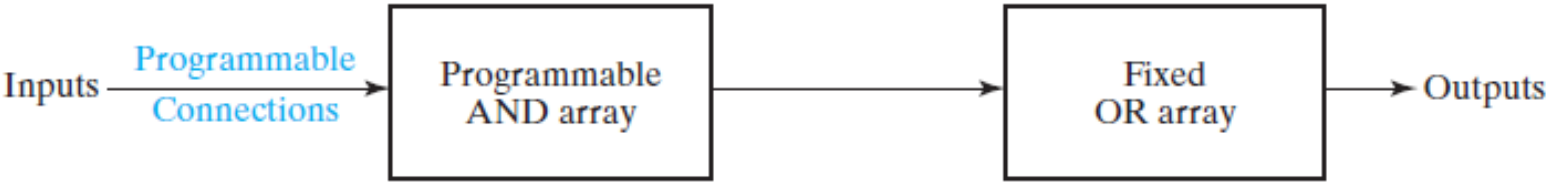


- Input (5 bit di indirizzo) va in ingresso a un **decoder** 5-to-32
- In uscita al decoder ci sono i 32 possibili indirizzi dei dati memorizzati
- Le uscite del decoder sono collegate a 8 **porte OR** a 32 ingressi **con connessioni programmabili** (le x indicano la presenza di una connessione)
- Nell'esempio, all'indirizzo 1 della memoria è contenuta la word «10010011»

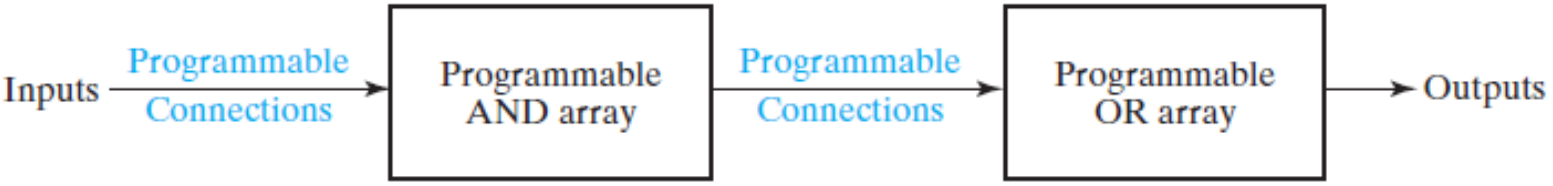
Programmable Logic Devices (PLD)



(a) Programmable read-only memory (PROM)



(b) Programmable array logic (PAL) device

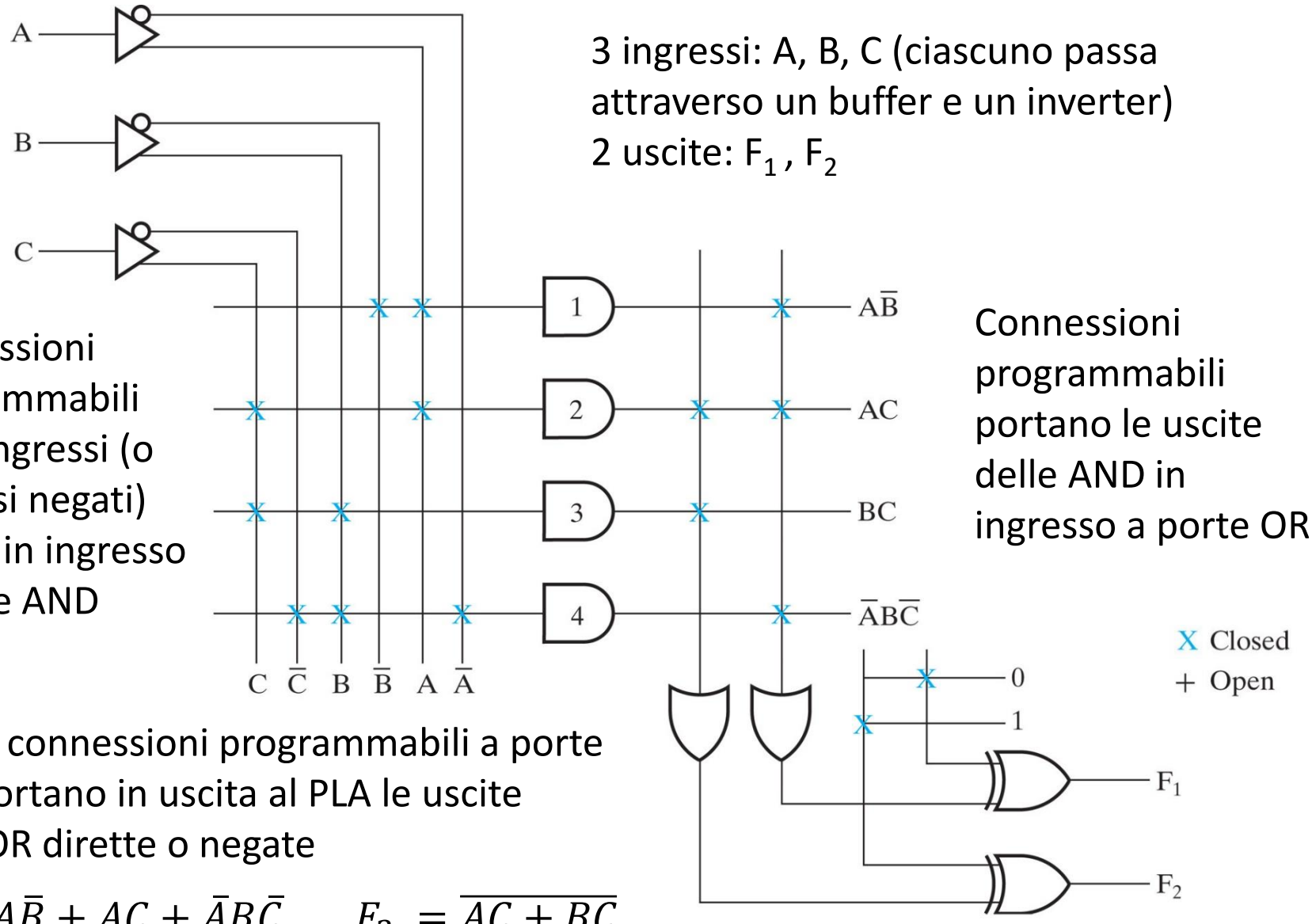


(c) Programmable logic array (PLA) device

Programmable Logic Array (PLA)

- E’ un PLD simile ad una ROM, ma il decoder è sostituito da una serie di **porte AND programmabili che generano i termini prodotto delle variabili di ingresso**
- I termini prodotto sono poi **collegati in modo selettivo a porte OR**, in modo analogo ad una ROM, per realizzare la somma dei minterm in uscita alle porte AND
- La differenza con una ROM è che non genera tutti i minterm corrispondenti alle variabili di ingresso, ma solo un sottoinsieme programmabile di questi
- I PLA commerciali contengono un numero molto elevato di porte e connessioni (altrimenti non sarebbero vantaggiosi dal punto di vista del costo)

PLA: esempio



3 ingressi: A, B, C (ciascuno passa attraverso un buffer e un inverter)
2 uscite: F_1 , F_2

Connessioni programmabili dagli ingressi (o ingressi negati) vanno in ingresso a porte AND

Connessioni programmabili portano le uscite delle AND in ingresso a porte OR

Infine, connessioni programmabili a porte XOR portano in uscita al PLA le uscite delle OR dirette o negate

$$F_1 = A\bar{B} + AC + \bar{A}B\bar{C} \qquad F_2 = \overline{AC + BC}$$

PLA: implementazione di un circuito combinatorio

- Le informazioni che servono per programmare un PLA affinché implementi una data funzione logica sono
  - Quali sono le connessioni tra gli ingressi (diretti o negati) e le porte AND
  - Quali sono le connessioni tra le porte AND e le porte OR
  - Se la somma di prodotti finale è diretta o negata
- Per ottenere un PLA con dimensioni minime, è importante **minimizzare** (tramite semplificazioni booleane) **il numero di termini prodotto**, cioè condividere i termini prodotto tra le uscite
  - Il numero di letterali all'interno di ciascun minterm è invece meno rilevante, essendo disponibili tutte le variabili di ingresso, dirette e negate
- Per la progettazione di circuiti combinatori tramite PLA si usano algoritmi di **ottimizzazione delle funzioni a due livelli su uscite multiple**

PLA: es. di implementazione di circuito

- Implementare le seguenti funzioni logiche tramite un PLA

$$F_1(A,B,C) = \sum m(3,5,6,7)$$

$$F_2(A,B,C) = \sum m(1,2,3,7)$$

PLA: es. di implementazione di circuito

$$F_1(A,B,C) = \sum m(3,5,6,7) \qquad F_2(A,B,C) = \sum m(1,2,3,7)$$

A \ B C	0 0	0 1	1 1	1 0
0	0	0	1	0
1	0	1	1	1

$$F_1(A,B,C) = BC + AC + AB$$

5 termini prodotto
(1 termine condiviso
tra F_1 e F_2)

A \ B C	0 0	0 1	1 1	1 0
0	0	1	1	1
1	0	0	1	0

$$F_2(A,B,C) = BC + \bar{A}B + \bar{A}C$$

PLA: es. di implementazione di circuito

$$F_1(A,B,C) = \sum m(3,5,6,7) \qquad F_2(A,B,C) = \sum m(1,2,3,7)$$

A \ B C	0 0	0 1	1 1	1 0
0	0	0	1	0
1	0	1	1	1

Copertura di F_1 negata:

$$\overline{F_1(A,B,C)} = \bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

$$F_1(A,B,C) = \overline{\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}}$$

4 termini prodotto (2 termini condivisi tra F_1 e F_2)

A \ B C	0 0	0 1	1 1	1 0
0	0	1	1	1
1	0	0	1	0

Copertura di F_2 diretta:

$$F_2(A,B,C) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + BC$$

=> Usando due implicanti non primi **ottimizziamo la condivisione** tra F_1 e F_2

Programmable Array Logic (PAL)

- E' un PLD con **una matrice programmabile di porte AND e una matrice fissa di porte OR**
- Il PAL è meno flessibile del PLA (dove entrambe le matrici di porte OR e AND sono programmabili), ma ha una struttura meno complessa
  - A differenza del PAL, non si possono condividere i termini prodotto (porte AND) tra diverse porte OR
- Per la progettazione di circuiti combinatori tramite PAL si usano algoritmi di **ottimizzazione delle funzioni a due livelli su singola uscita**
- Tipicamente un PAL commerciale contiene almeno 8 ingressi e 8 uscite, altrimenti non risulterebbe vantaggioso dal punto di vista dei costi

PAL: es. di implementazione di circuito

- Implementare le seguenti funzioni logiche tramite un PAL

$W(A,B,C,D) = \sum m\,(2,12,13)$

$X(A,B,C,D) = \sum m\,(7,8,9,10,11,12,13,14,15)$

$Y(A,B,C,D) = \sum m\,(0,2,3,4,5,6,7,8,10,11,15)$

$Z(A,B,C,D) = \sum m\,(1,2,8,12,13)$

PAL: es. di implementazione di circuito

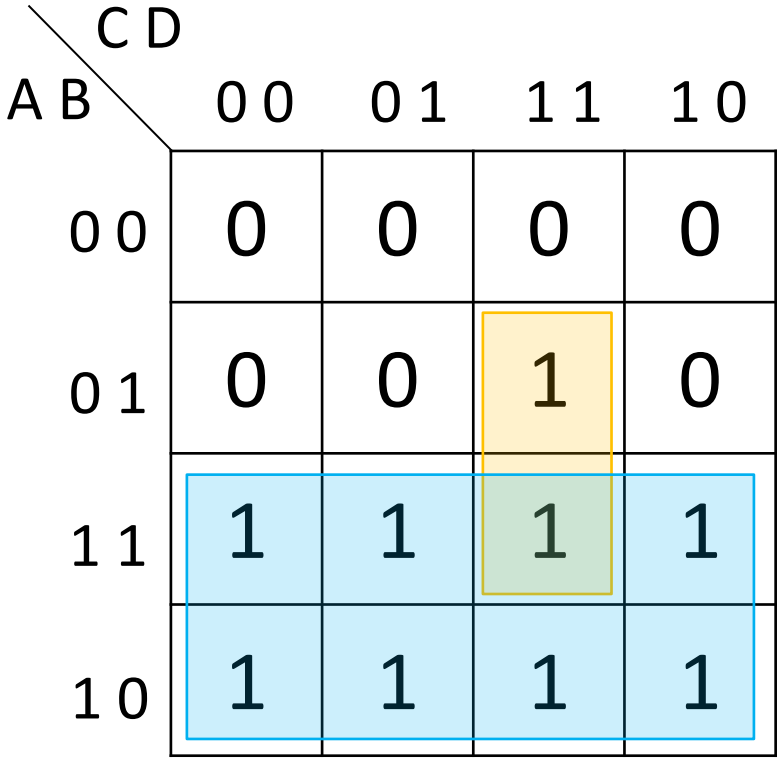
$$W(A,B,C,D) = \sum m\,(2,12,13)$$

A B \ C D	0 0	0 1	1 1	1 0
0 0	0	0	0	1
0 1	0	0	0	0
1 1	1	1	0	0
1 0	0	0	0	0

$$W = AB\bar{C} + \bar{A}\bar{B}C\bar{D}$$

PAL: es. di implementazione di circuito

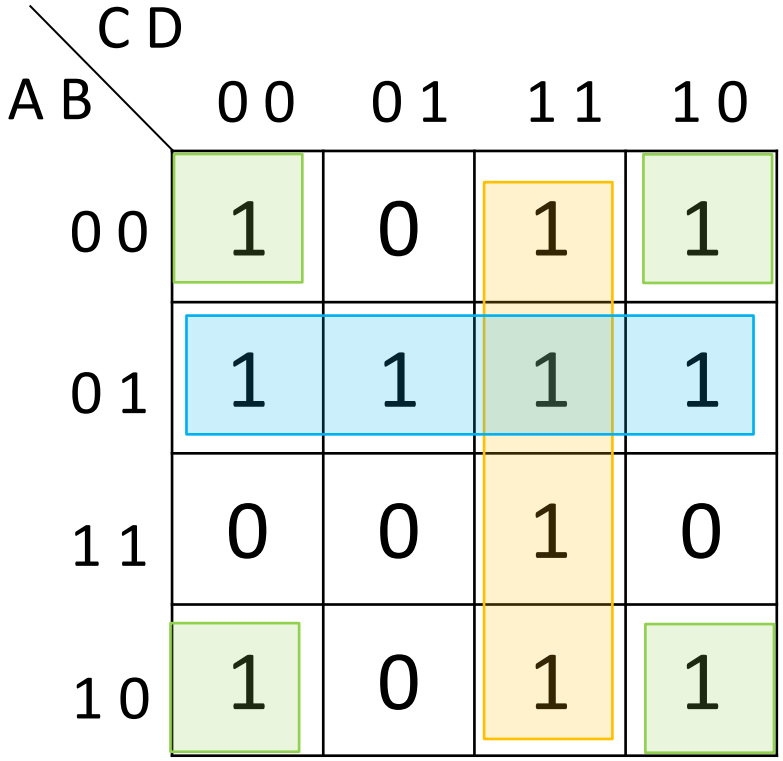
$$X(A,B,C,D) = \sum m\,(7,8,9,10,11,12,13,14,15)$$



$$X = A + BCD$$

PAL: es. di implementazione di circuito

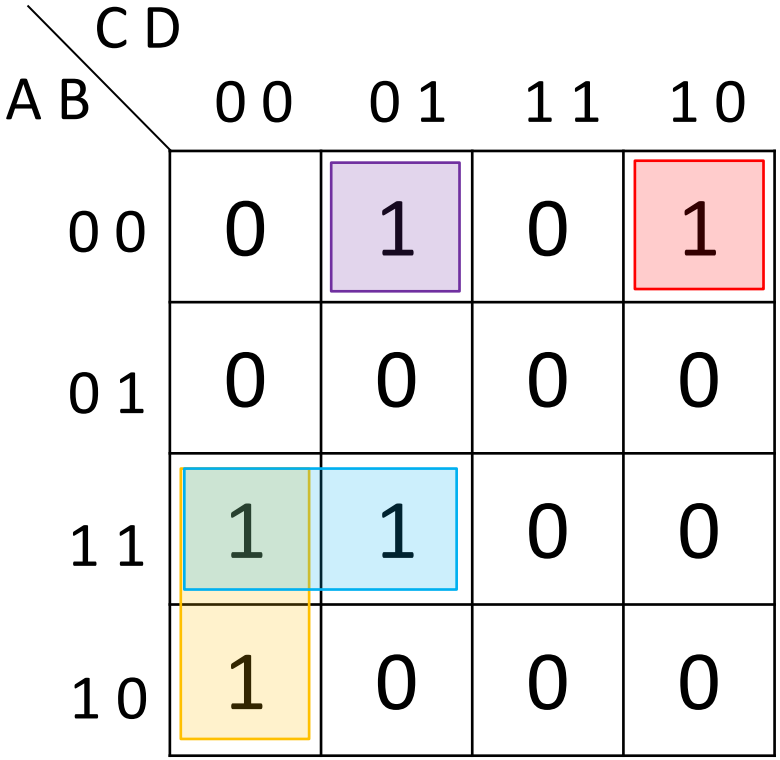
$$Y(A,B,C,D) = \sum m\,(0,2,3,4,5,6,7,8,10,11,15)$$



$$Y = \bar{A}B + CD + \bar{B}\bar{D}$$

PAL: es. di implementazione di circuito

$$Z(A,B,C,D) = \sum m\,(1,2,8,12,13)$$



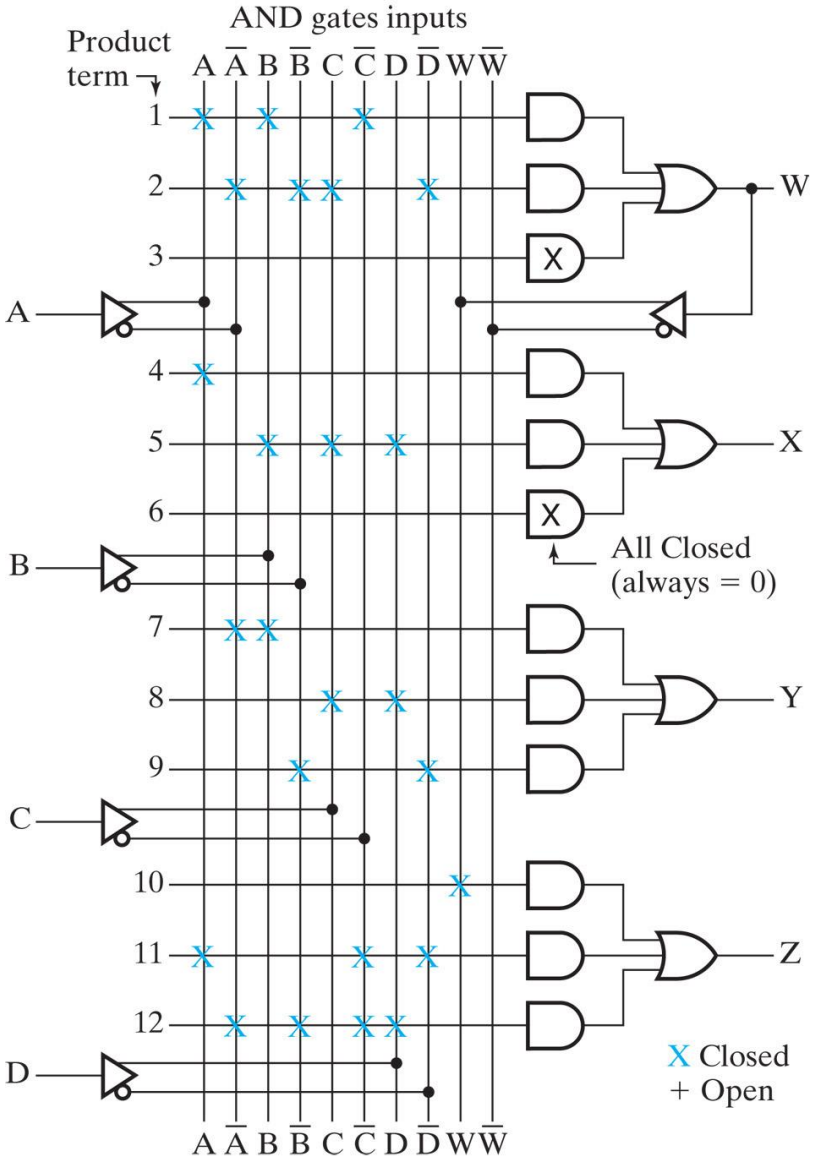
A B \ C D	0 0	0 1	1 1	1 0
0 0	0	1	0	1
0 1	0	0	0	0
1 1	1	1	0	0
1 0	1	0	0	0

$$Z = AB\bar{C} + A\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D}$$

$$W = AB\bar{C} + \bar{A}\bar{B}C\bar{D}$$

$$\Rightarrow Z = W + A\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D$$

PAL: es. di implementazione di circuito



$$W = AB\bar{C} + \bar{A}\bar{B}C\bar{D}$$

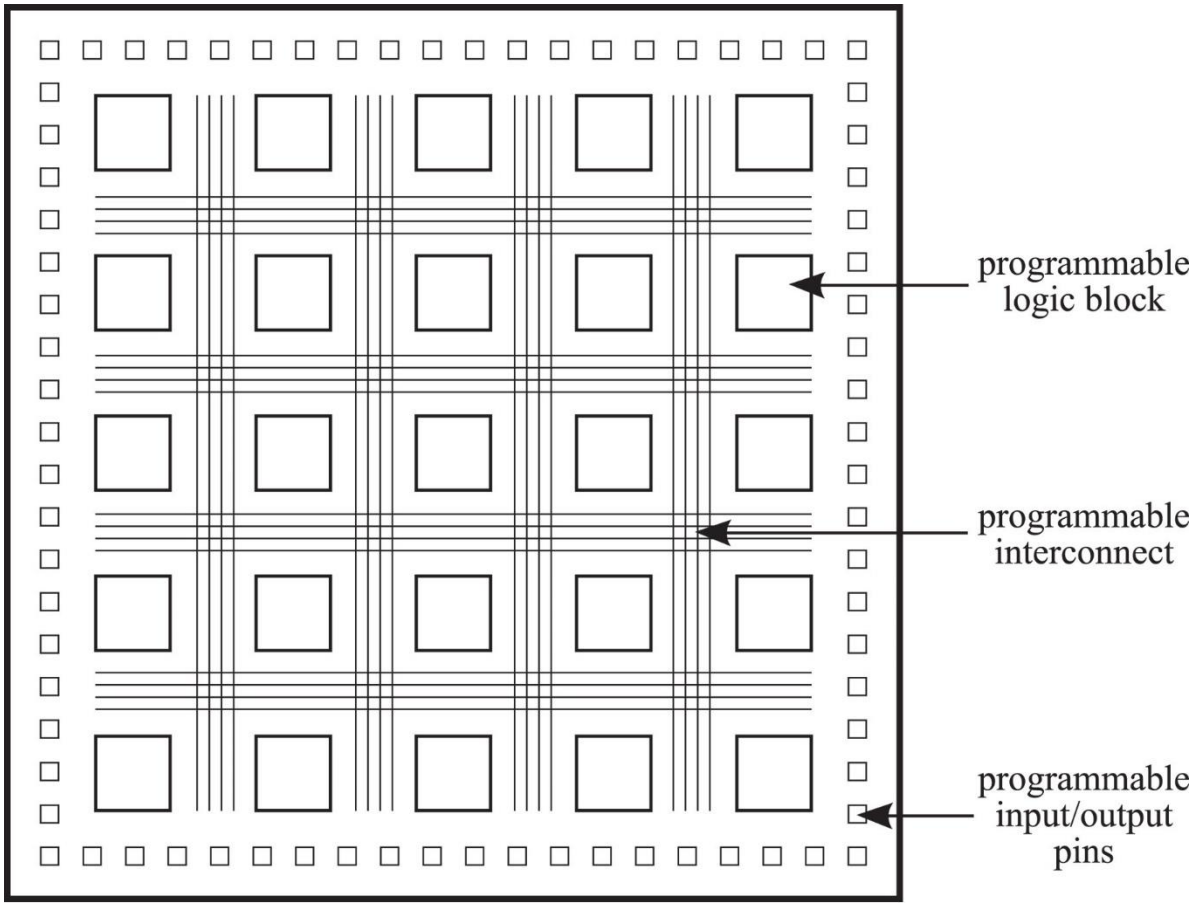
$$X = A + BCD$$

$$Y = \bar{A}B + CD + \bar{B}\bar{D}$$

$$Z = W + A\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D$$

Field Programmable Gate Array (FPGA)

- Gli FPGA sono oggi i PLD più diffusi
- Le caratteristiche che accomunano le diverse soluzioni nel mercato sono la presenza di
  - **Blocchi logici programmabili**
  - **Interconnessioni programmabili**
  - **Pin di I/O programmabili**

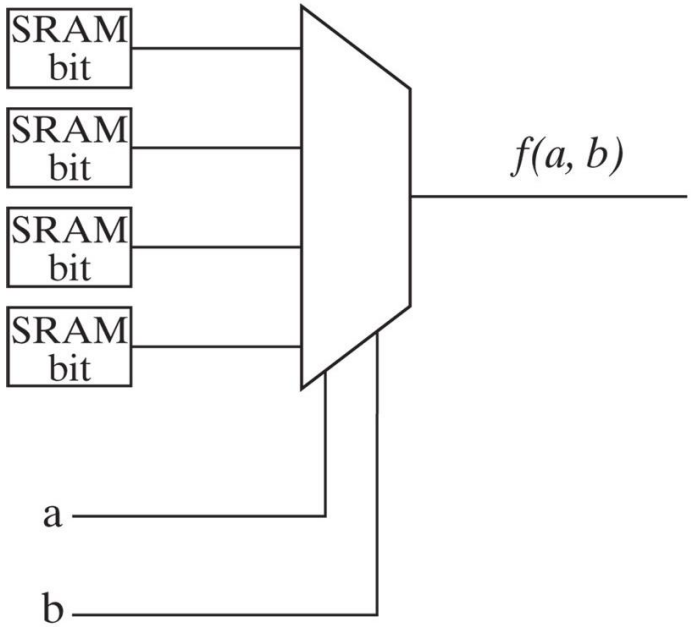


Field Programmable Gate Array (FPGA)

- In aggiunta a questi blocchi configurabili, gli FPGA possono contenere **blocchi specializzati di logica dedicata** (memorie, componenti per il calcolo aritmetico, microprocessori)
- I vantaggi di usare gli FPGA rispetto ad altri dispositivi programmabili sono la **disponibilità di blocchi logici configurabili e flip-flop**, insieme alla **facilità di riconfigurazione**
- Le informazioni sul circuito implementato sono contenute nella **memoria di configurazione** che può essere volatile (SRAM) oppure non volatile (Flash, fuse/anti-fuse, …)
- Per programmare un FPGA bisogna specificare il valore dei bit di configurazione per i blocchi logici, routing (interconnessioni) e pin di ingresso/uscita

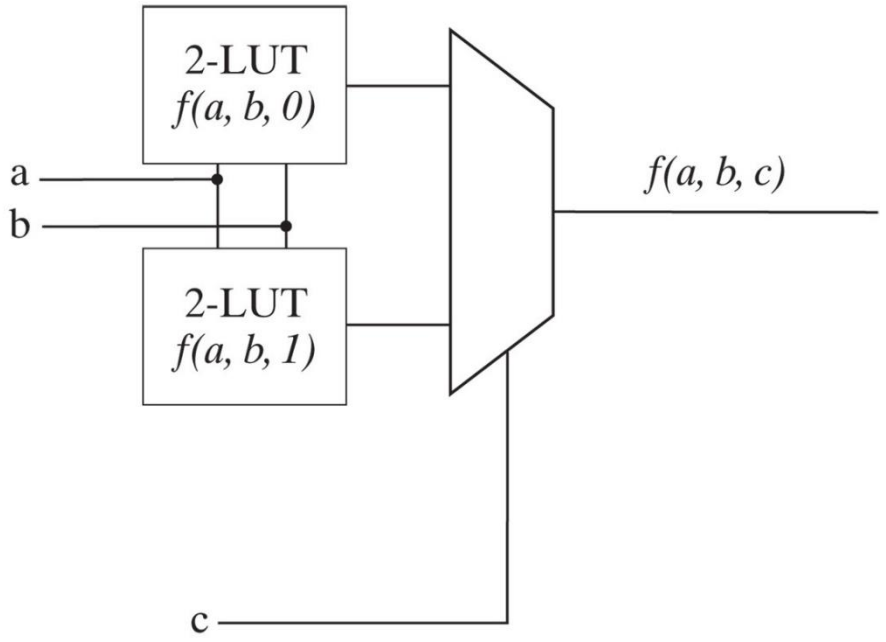
Blocchi logici programmabili

- Negli FPGA sono disponibili blocchi con logica combinatoria e sequenziale, configurabili dall'utente per svolgere le funzioni desiderate
- In molte famiglie di FPGA i blocchi logici programmabili contengono **Look-Up Table (LUT)**, cioè memorie di dimensione 2^k x 1 che implementano la **tabella di verità per una funzione combinatoria di k variabili**



LUT a 2 ingressi (a,b):
Ponendo opportunamente a '0' o '1' i bit della memoria di configurazione dell'FPGA (bit in ingresso al mux), possiamo implementare la funzione logica desiderata, tra le possibili 16 funzioni booleane a 2 variabili

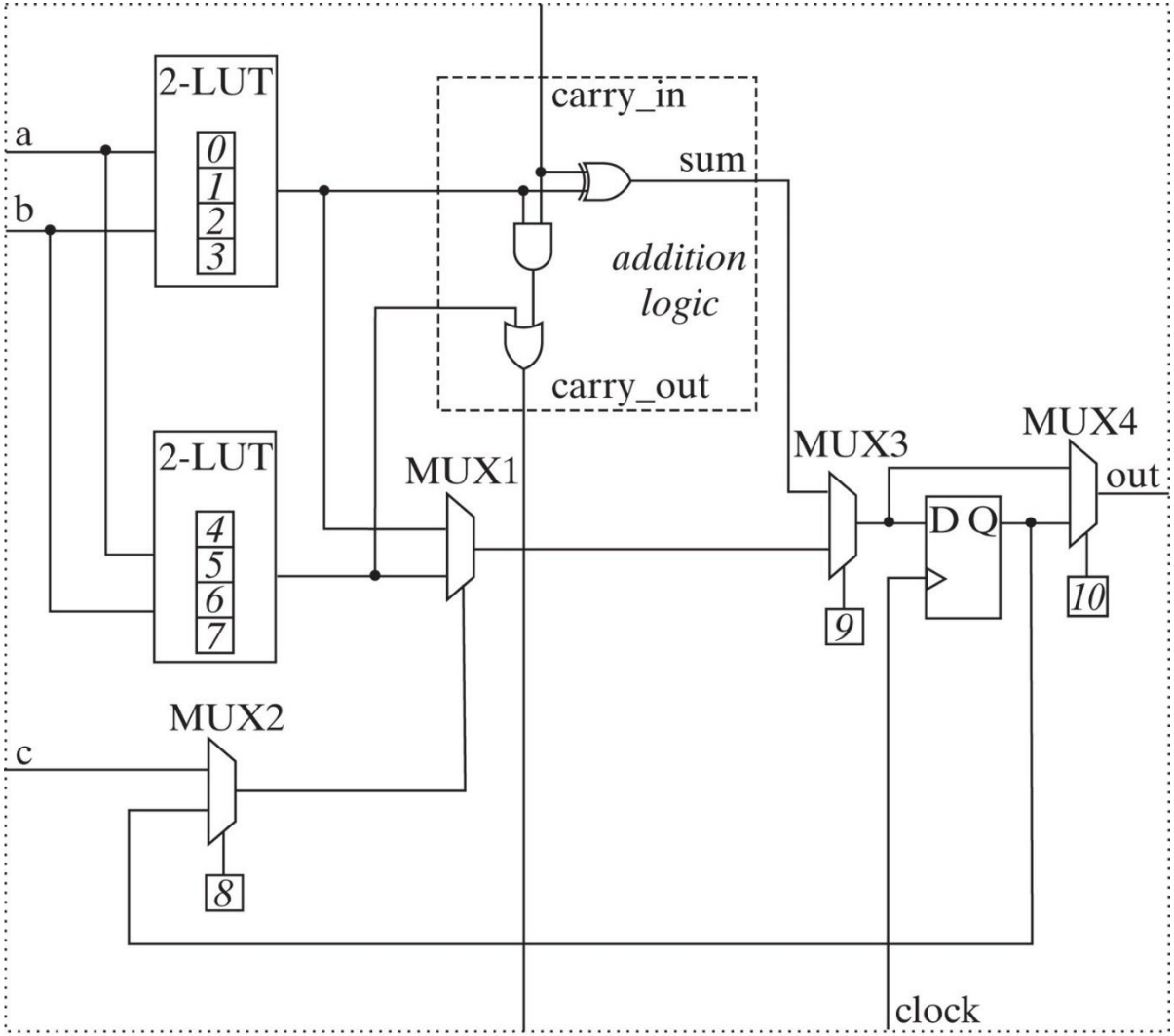
Blocchi logici programmabili



LUT a 3 ingressi (a,b,c): Connettendo due LUT a 2 ingressi con un multiplexer 2-to-1, otteniamo una LUT a 3 ingressi (a,b,c)

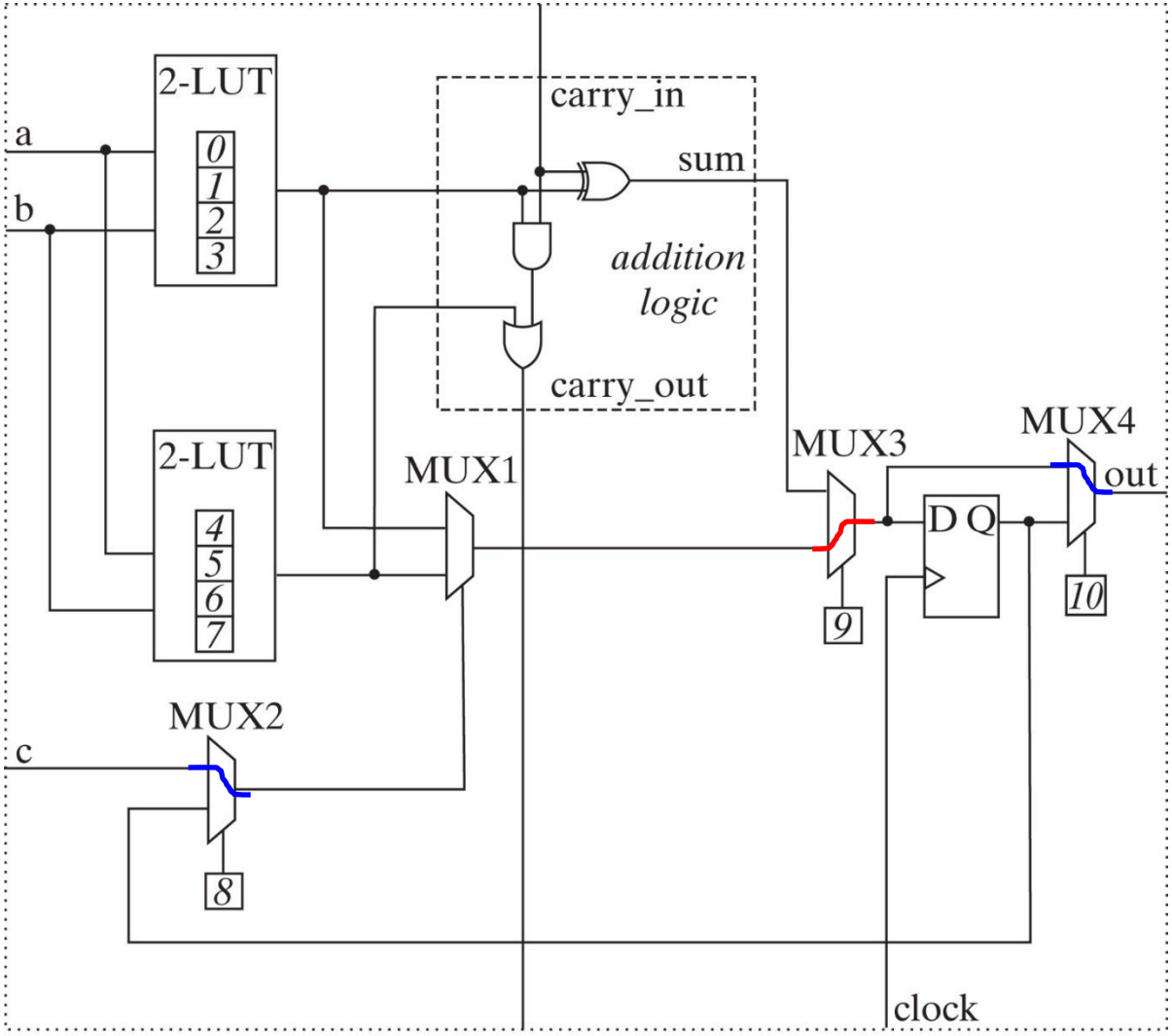
- Oltre alle LUT, i blocchi logici programmabili contengono anche **altri elementi** (**flip-flop, multiplexer, blocchi artimetici**, …) per implementare le più svariate funzioni logiche

Blocchi logici programmabili: esempio



- MUX1 + 2-LUT: implementano una funzione combinatoria di 3 variabili f(a, b, x), i config. bit 0-7 determinano la tabella di verità della funzione
- MUX2 (config. bit 8) seleziona se la terza variabile di ingresso di f è c o l'uscita del FF
- MUX3 (config. bit 9) seleziona uno tra l'uscita di MUX 1 e il bit sum del full adder come ingresso del FF
- MUX4 (config. bit 10) seleziona uno tra il blocco combinatorio e l'uscita del FF come uscita del blocco programmabile

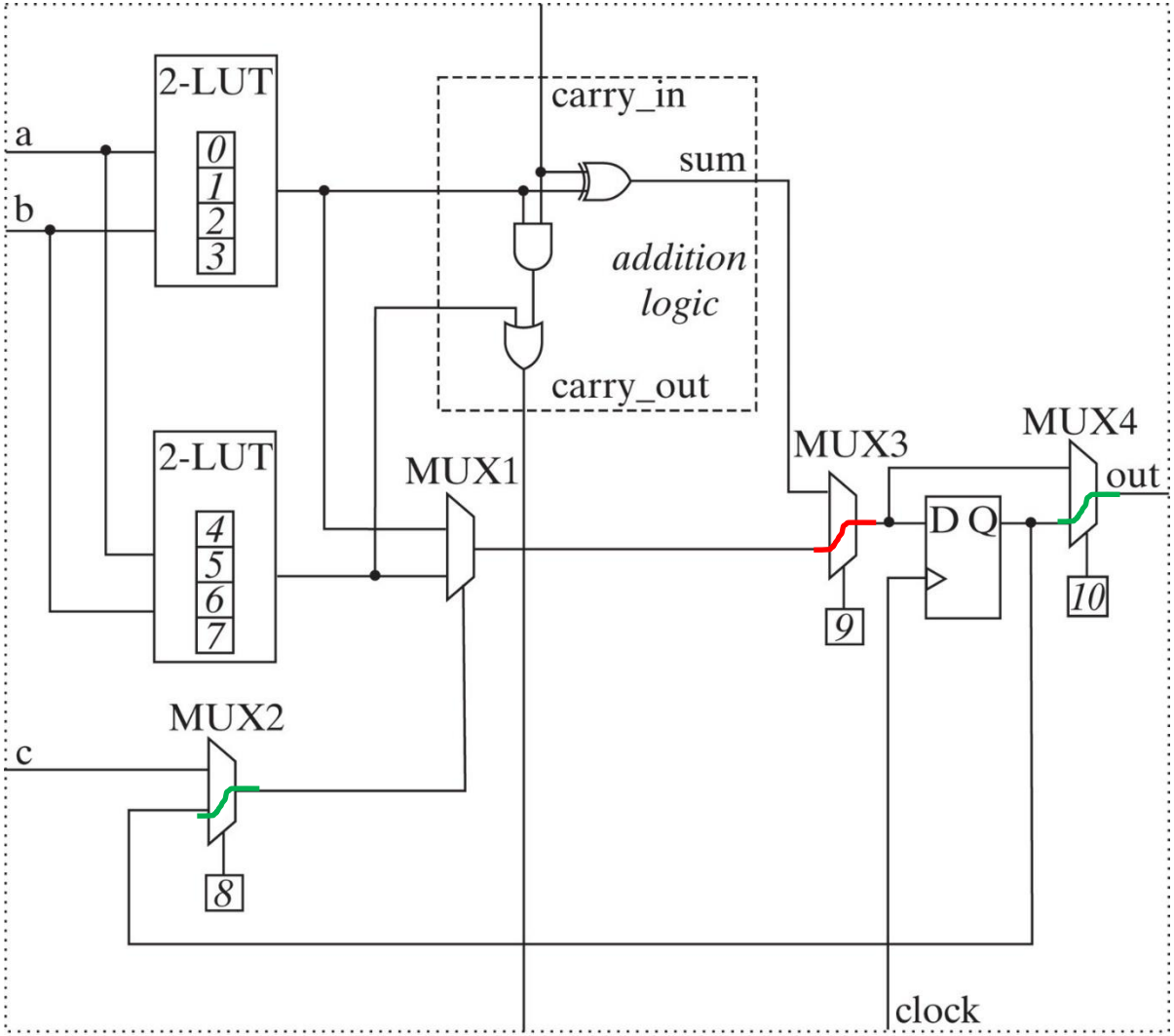
Blocchi logici programmabili: esempio



Supponiamo che MUX3 (config. bit 9) sia programmato in modo da selezionare l'uscita di MUX1. A seconda di come sono configurati MUX2 e MUX4, l'uscita out può essere per esempio

- Una qualsiasi funzione combinatoria di a, b, c (se config. bit 8 e 10 = '0')

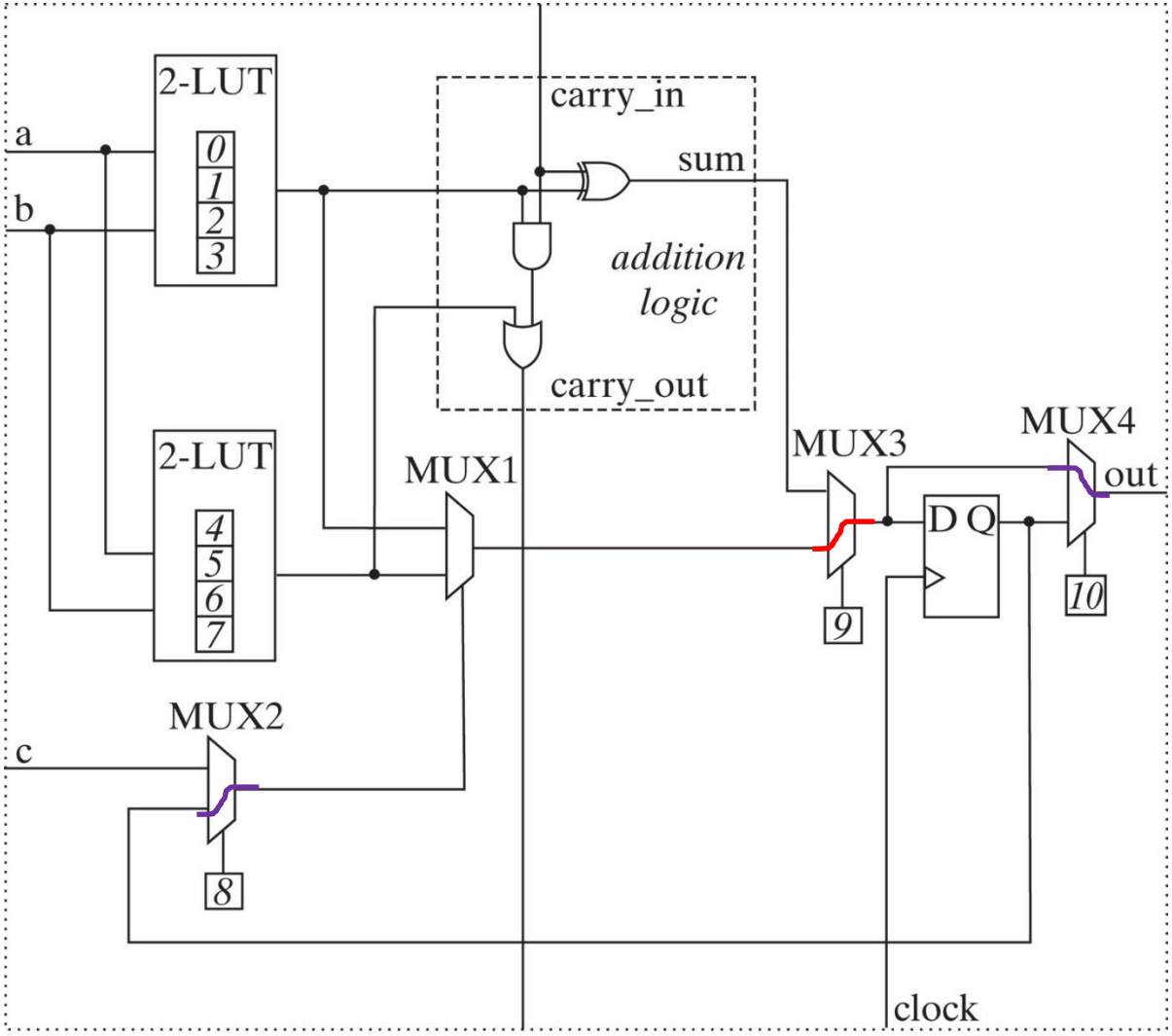
Blocchi logici programmabili: esempio



Supponiamo che MUX3 (config. bit 9) sia programmato in modo da selezionare l'uscita di MUX1. A seconda di come sono configurati MUX2 e MUX4, l'uscita out può essere per esempio

- Una macchina di Moore (se bit di config. 8 e 10 = '1', l'uscita dipende solo dallo stato presente e non dagli ingressi)

Blocchi logici programmabili: esempio



Supponiamo che MUX3 (config. bit 9) sia programmato in modo da selezionare l'uscita di MUX1. A seconda di come sono configurati MUX2 e MUX4, l'uscita out può essere per esempio

- Una macchina di Mealy (se bit di config. 8 = '1' e 10 = '0', l'uscita dipende non solo dallo stato, ma anche dagli ingressi)

Disclaimer

Figures from *Logic and Computer Design Fundamentals*, Fifth Edition, GE Mano | Kime | Martin