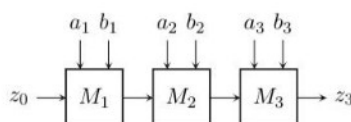


COMPARATORI

È UN CIRCUITO CHE CONFRONTA DUE PAROLE BINARIE A e B .
IN USCITA AURÒ UN BIT CHE MI DIRÀ SE I NUMERI SONO UGUALI O NO: VARIABILE BOOLEANA.

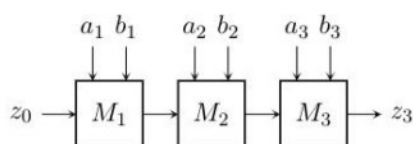
- Il risultato di un comparatore determina se $A = B$
- Il confronto può essere effettuato su ciascuna coppia di bit in moduli separati
- Tuttavia, è necessario *propagare* il risultato della comparazione dai moduli precedenti



↓
ciascun modulo
confronta due singoli bit.

↓
quindi ~~dentro~~ c'è un circuito che mi dice se
i due bit sono uguali o diversi. Questa informazione
verrà propagata ai moduli successivi.

APPENA TROVO DEI BIT DIFFERENTI, ESCE 0 DAL MODULO, E QUESTO 0 SI PROPAGHERÀ FINO ALL'USCITA.

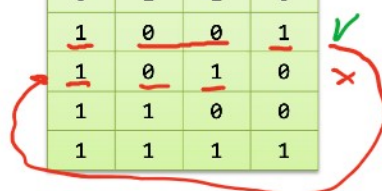


- L'uscita è 1 se e solo se riceviamo un 1 dagli stadi precedenti (tutti i bit precedenti sono uguali) e se i due bit analizzati nel modulo corrente sono uguali

- I mintermini della funzione sono soltanto due:

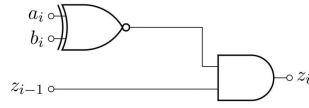
$$z_i = z_{i-1} \bar{a} \bar{b} + z_{i-1} a b = z_{i-1} (a \odot b)$$

z_{i-1}	a_i	b_i	z_i
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



ZOOM SU CIASCUN MODULO

- La realizzazione circuitale del modulo M_i è immediata:



- È interessante però realizzare un comparatore che possa discriminare:
 - $A = B$: codifica di output 00
 - $A > B$: codifica 10
 - $A < B$: codifica 01

MA IL CRITICAL PATH È LUNGO ALMENO PER IL NUMERO DI CIFRE CHE DEVO CONTROLLARE: ALLUNGO IL CLOCK.

ESERCIZIO DA FARE

- Esercizio:** mostrare che, data le configurazioni ammissibili, le espressioni minime per calcolare le uscite del modulo M_i corrispondono a:

- $z_{i,a} = z_{i-1,a}(a_i + \overline{b_i}) + a_i \overline{b_i}$
- $z_{i,b} = z_{i-1,b}(b_i + \overline{a_i}) + b_i \overline{a_i}$

