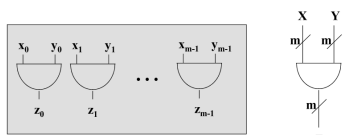


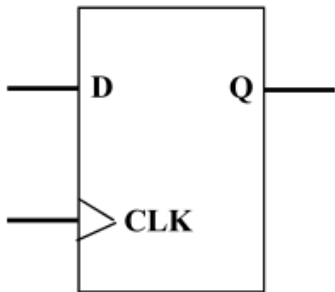
$\{x_0, \dots, x_m\}$ l'unita di dato manipolato è la **parola**

i componenti usati nei progetti sono blocchi per la **memo.** e la **manipolazione** di parole



Progettazione a livello di registro

un modo composto da PL che permette di memorizzare



la versione più comune e quella di tipo D

Q mostra il valore memorizzato

Ad ogni "botta" di clock il FF memorizza il valore presente in D

Flip Flop

Progettazione di circuiti

circuiti in cui i valori delle uscite dipendono sia dai valori applicati in quell'istante sia in quelli successivi

il circuito a memoria

implementano una **FSM** (Finite State Machine o Macchina a Stati Finiti)

Circuiti Sequenziali

circuiti in cui i **valori delle uscite** dipendono **esclusivamente** dai valori applicati sui suoi **ingressi** in quell'istante

Implementano funzioni Booleane

Circuiti combinatori

Tensione e bit

Convenzionalmente si assume che

una tensione **analoga** a quella di **alimentazione** corrisponda ad un valore logico

1

una tensione **analoga** a quella della **massa** corrisponda ad un valore logico

0

NOT



NOT

x	\bar{x}
0	1
1	0

$\bar{x} = x' = \neg x$

AND



AND

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

$x \wedge y = xy$

OR

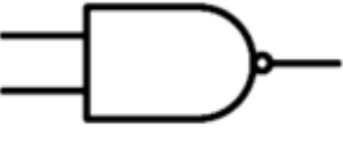


OR

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

$x \vee y = x + y$

NAND

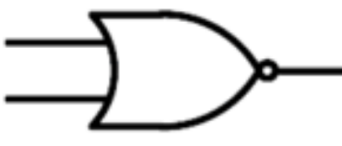


NAND

x	y	$\overline{x \wedge y}$
0	0	1
0	1	1
1	0	1
1	1	0

$\overline{x \wedge y} = \overline{xy}$

NOR

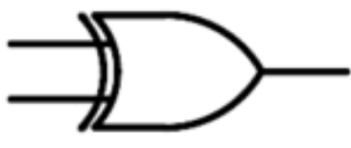


NOR

x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

$\overline{x \vee y} = \overline{x + y}$

XOR

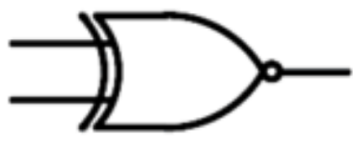


XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

$x \oplus y$

NXOR



NXOR

x	y	$\overline{x \oplus y}$
0	0	1
0	1	0
1	0	0
1	1	1

$\overline{x \oplus y}$