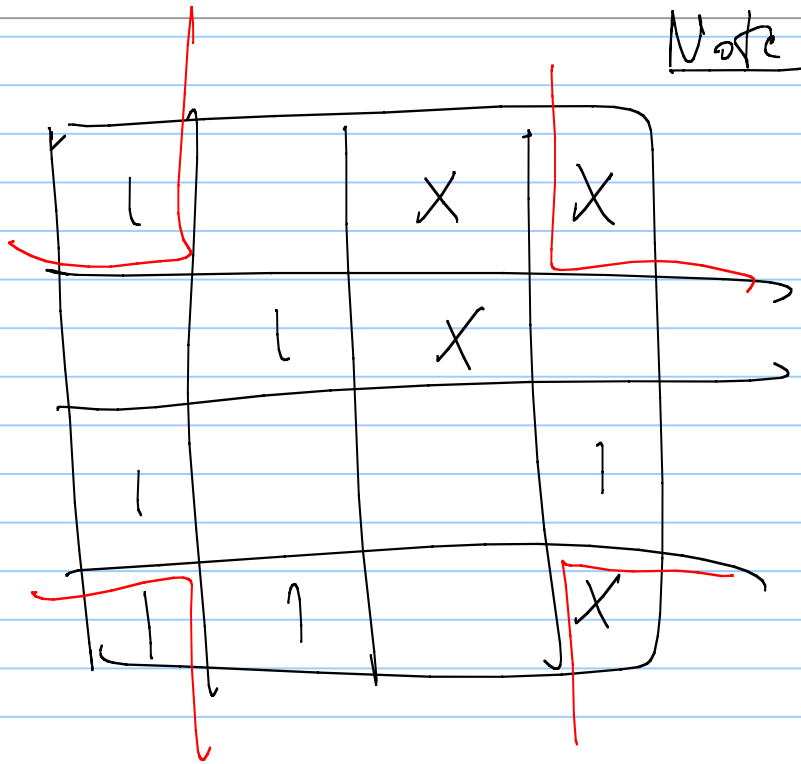
Note Title 4/20/2009



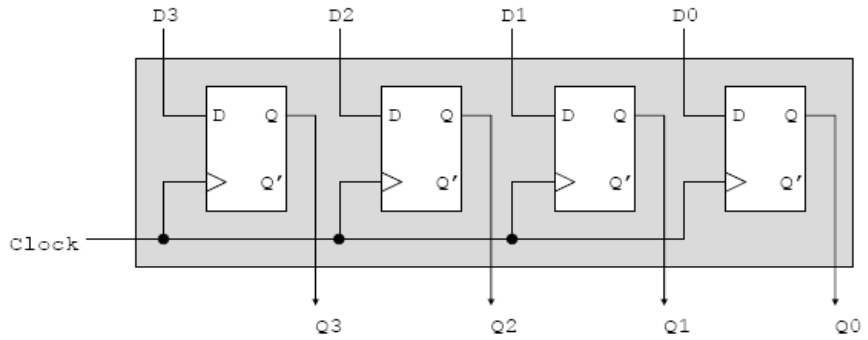
Note on Problem 1 of Take-home Test 2

$B'D'$ is an essential prime implicant.

Registers (ch. 12) are simple sequential circuits.
They are simple enough that the general techniques of
Chs 13-16 are not used to analyze and design them.
Registers - memorize a fixed amount of information
- they operate on their contents as follows
  - shift (right or left)
  - load (parallel or serial)
  - retrieve (read)

Register example images are from M.G. Soni's course
web site (mentioned before).

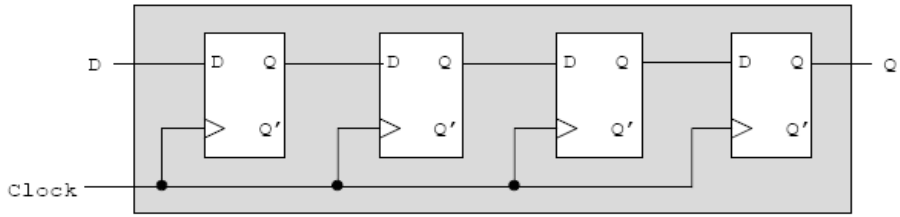
- Registro *parallelo-parallelo*
  - Esempio di registro a 4 bit.



A 4-bit Parallel-parallel register

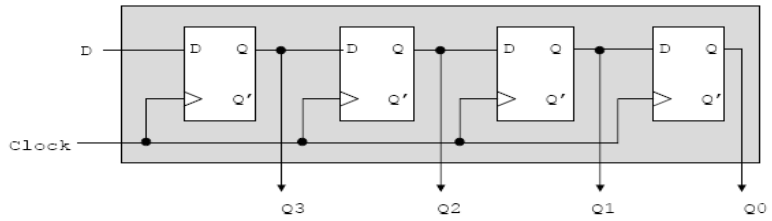
No shifting allowed

- Registro *serie-serie* (*Shift Register - Registro a Scorrimento*)
  - Esempio di registro a 4 bit



Shifts one bit to the right at every active clock signal, leftmost bit is replaced by the register input; rightmost bit is lost

- Registro *serie-parallelo*
  - Esempio di registro a 4 bit

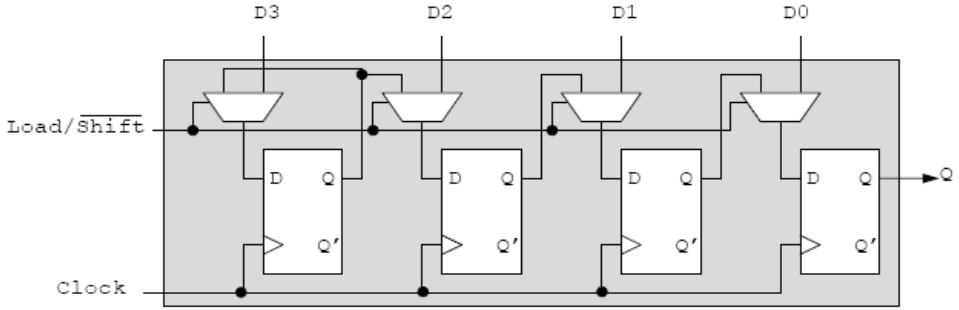


4-bit series-parallel register

Input is serial; output is parallel.

This is a shift register, with shifting done as in the previous register.

- Registro *parallelo-serie*
  - Esempio a 4 bit con scorrimento aritmetico (Shift Destro): In fase di traslazione, ricopia il bit più significativo nella posizione più significativa (estensione del segno)

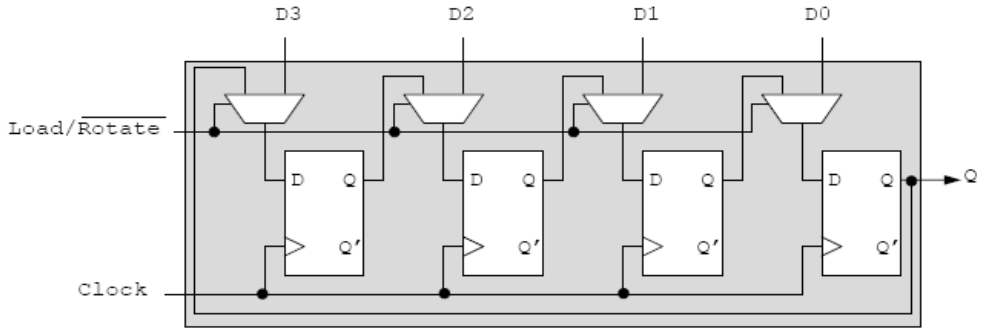


4-bit right-shift parallel-series register with persistent sign (MSB)

Note use of 2-to-1 MUX.

Note feedback line

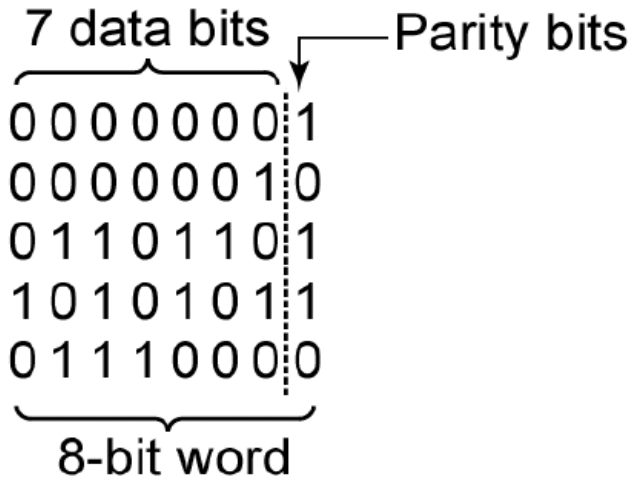
- Registro *circolare* a 4 bit
  - Esempio a 4 bit con rotazione a destra: In fase di traslazione, trasferisce il bit meno significativo al posto di quello più significativo, spostando i rimanenti di una posizione a destra.



4-bit circular register

Note feedback line.

Ch. 13: Analysis of Clocked Sequential Circuits



Section 13.1, p. 362

including the parity bit

Parity is odd when a correct string has an odd number of 1s

Parity is even when a correct string has an even number of 1s.

The resulting code can detect errors in one bit.

The example here is for an odd-parity code.

We are going to design a sequential parity checker

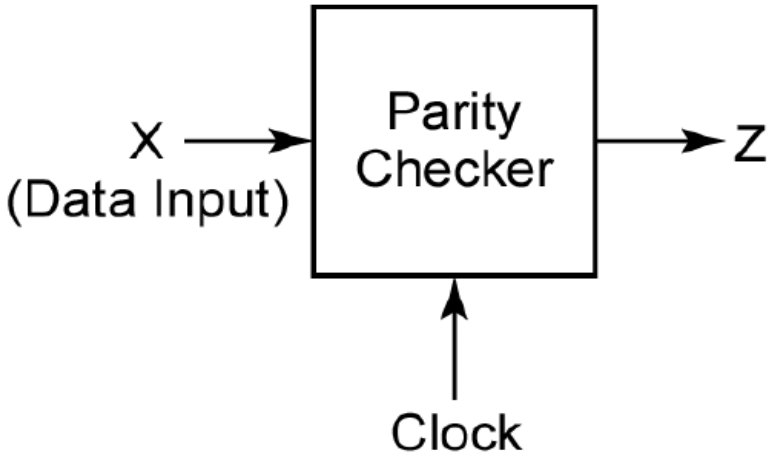


Figure 13-1: **Block Diagram for Parity Checker**

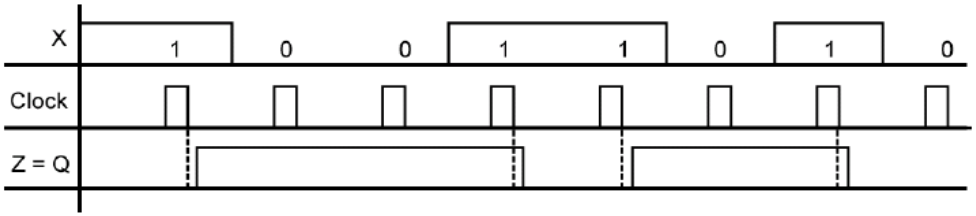


Figure 13-2: **Waveforms for Parity Checker**

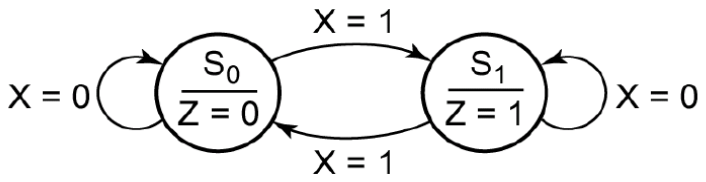


Figure 13-3: **State Graph for Parity Checker**

The output Z is a function of the state S (S_0 or S_1) only : this sequential circuit is a Moore machine.

Table 13-1: State Table for Parity Checker

(a)

Present State	Next State $X = 0$	Next State $X = 1$	Present Output
S_0	S_0	S_1	0
S_1	S_1	S_0	1

(b)

Q	Q^+ $X = 0$	Q^+ $X = 1$	T $X = 0$	T $X = 1$	Z
0	0	1	0	1	0
1	1	0	0	1	1

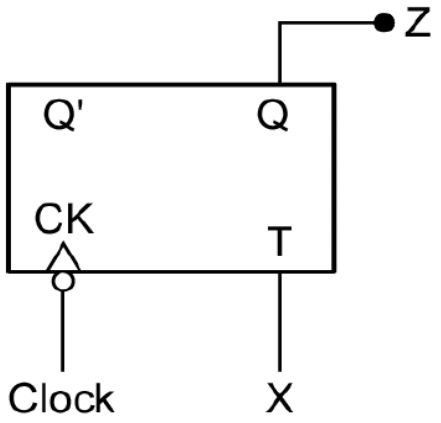


Figure 13-4: **Parity Checker**