

CS221: Digital Design

Register and Memory

A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- Register
 - Parallel Load, Parallel out : (PIPO)
 - Serial Load, Wrap around load, Serial out (SISO)
 - PISO, SIPO Register
- Multifunction Register : How to design?
- Memory Design using array of PIPO registers
 - RAM: Random Access Memory

Register

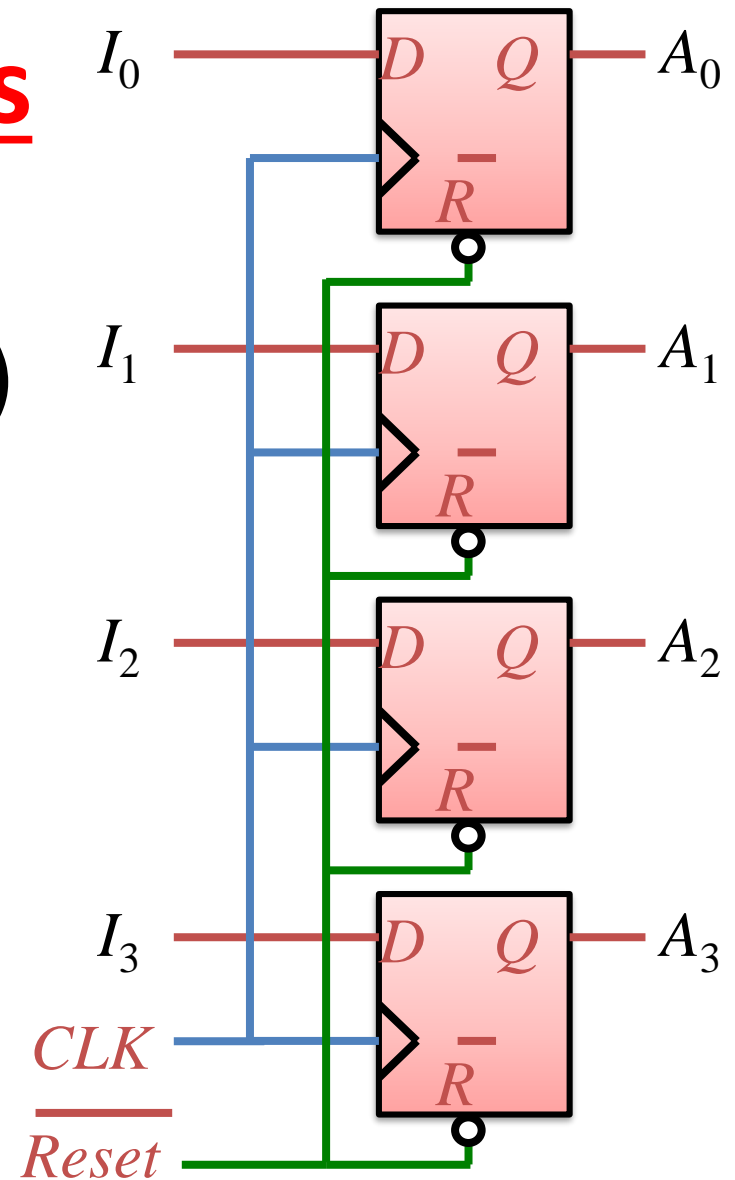
- Register
 - Parallel Load, Parallel out : (PIPO)
 - Serial Load, Wrap around load, Serial out (SISO)
 - PISO, SIPO Register
- Multifunction Register : How to design?
- Memory Design using array of PIPO registers

Important point in register

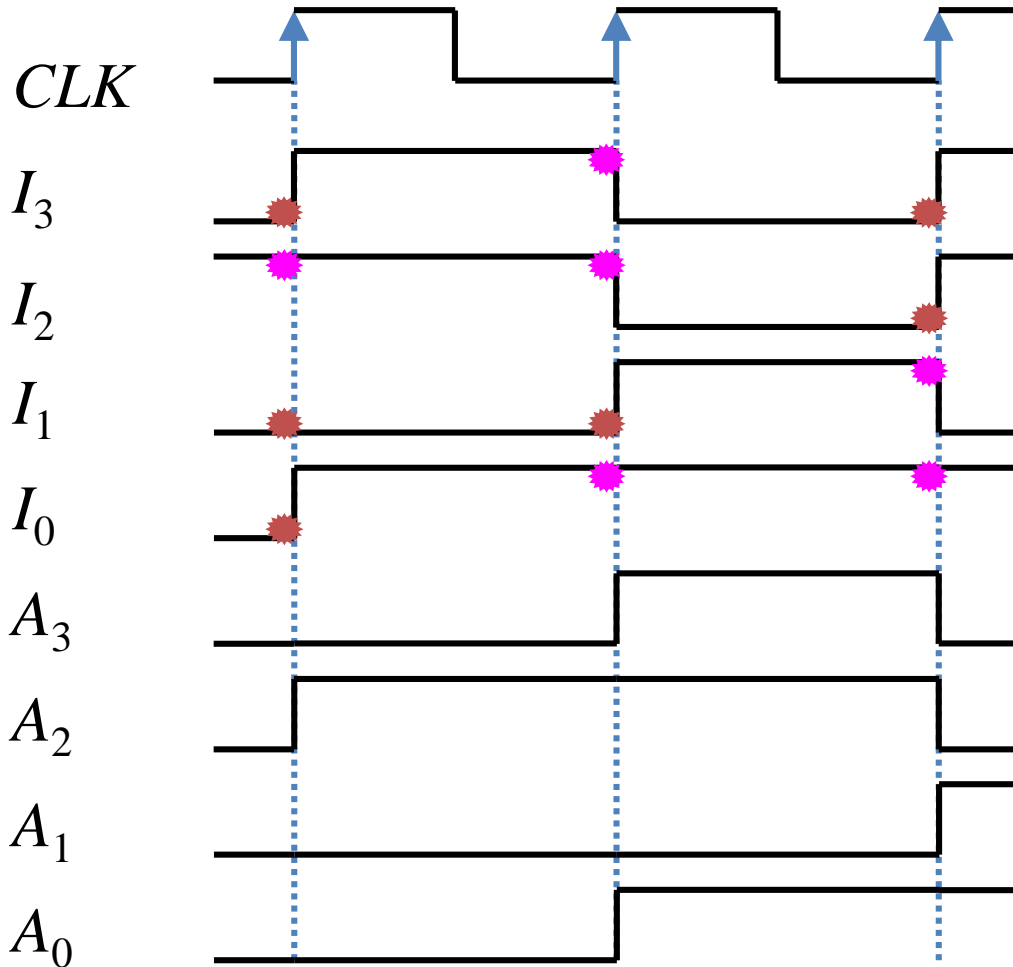
1. Input get reflected to output : Just before rising edges
2. Don't play with clock, it will add delay to clock and disturbed the other circuit
3. Use Mux for multiple inputs to D-FF of registers

Registers

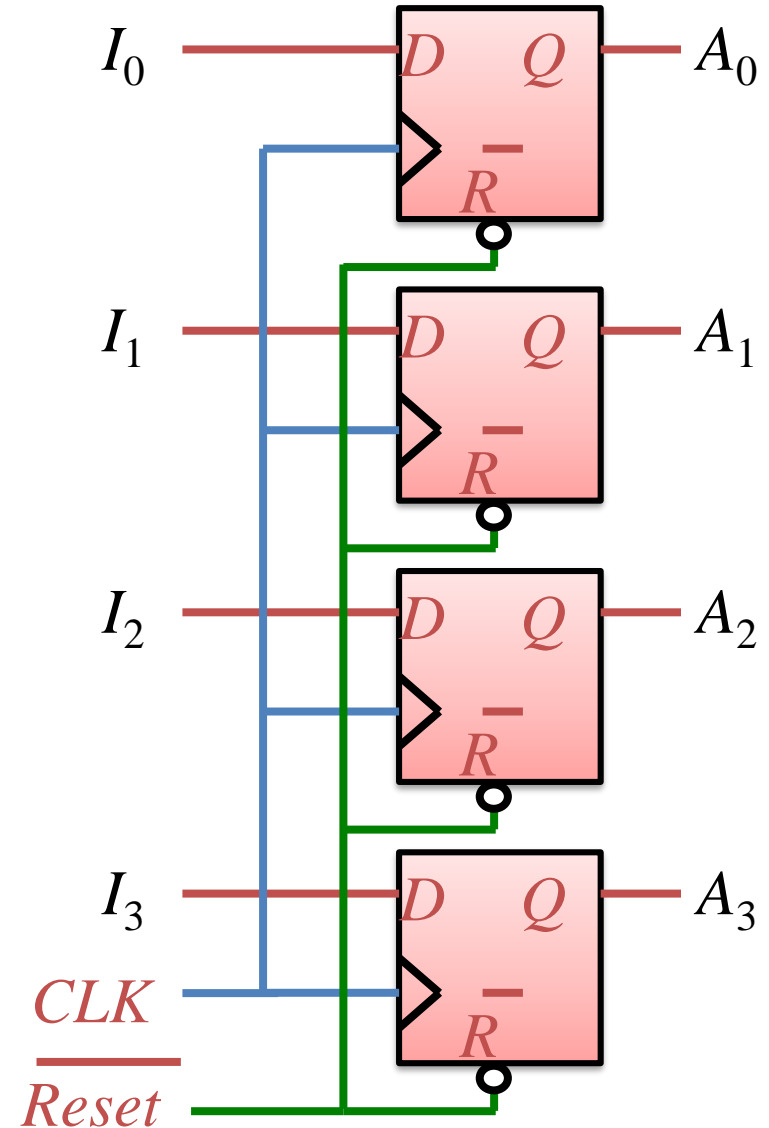
- Group of D Flip-Flops
- **Synchronized (Single Clock)**
- Store Data



Registers



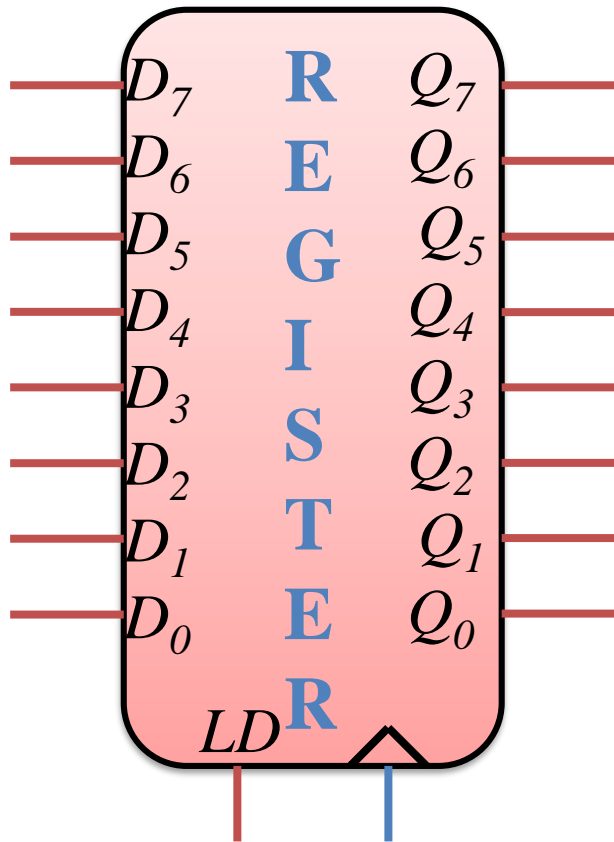
Note: New data has to go in with every clock



See carefully: Input at the dotted will be reflected to output : Just before rising edges

Registers with Parallel Load

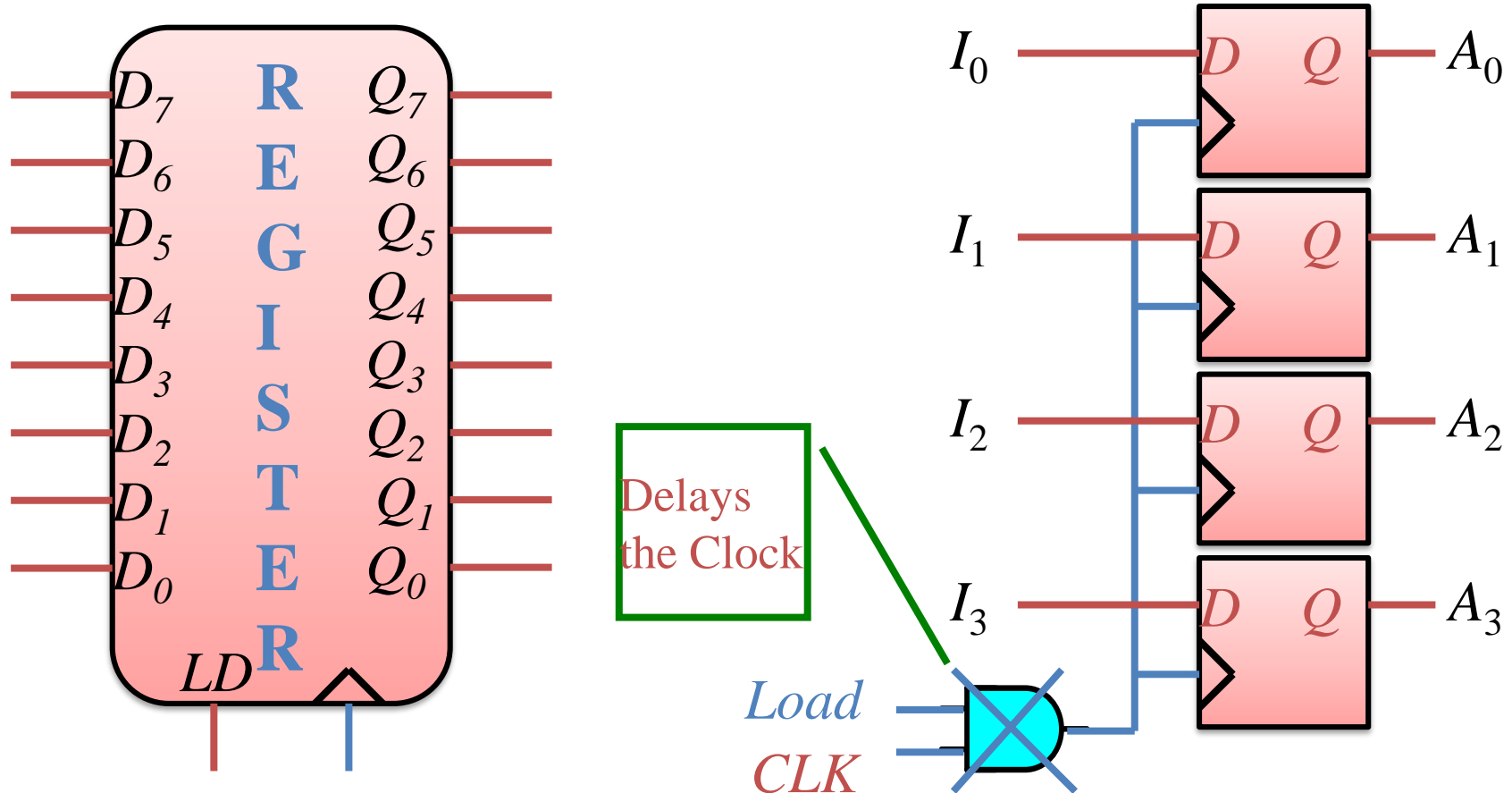
- Control **Loading** the Register with New Data



LD	$Q(t+1)$
0	$Q(t)$
1	D

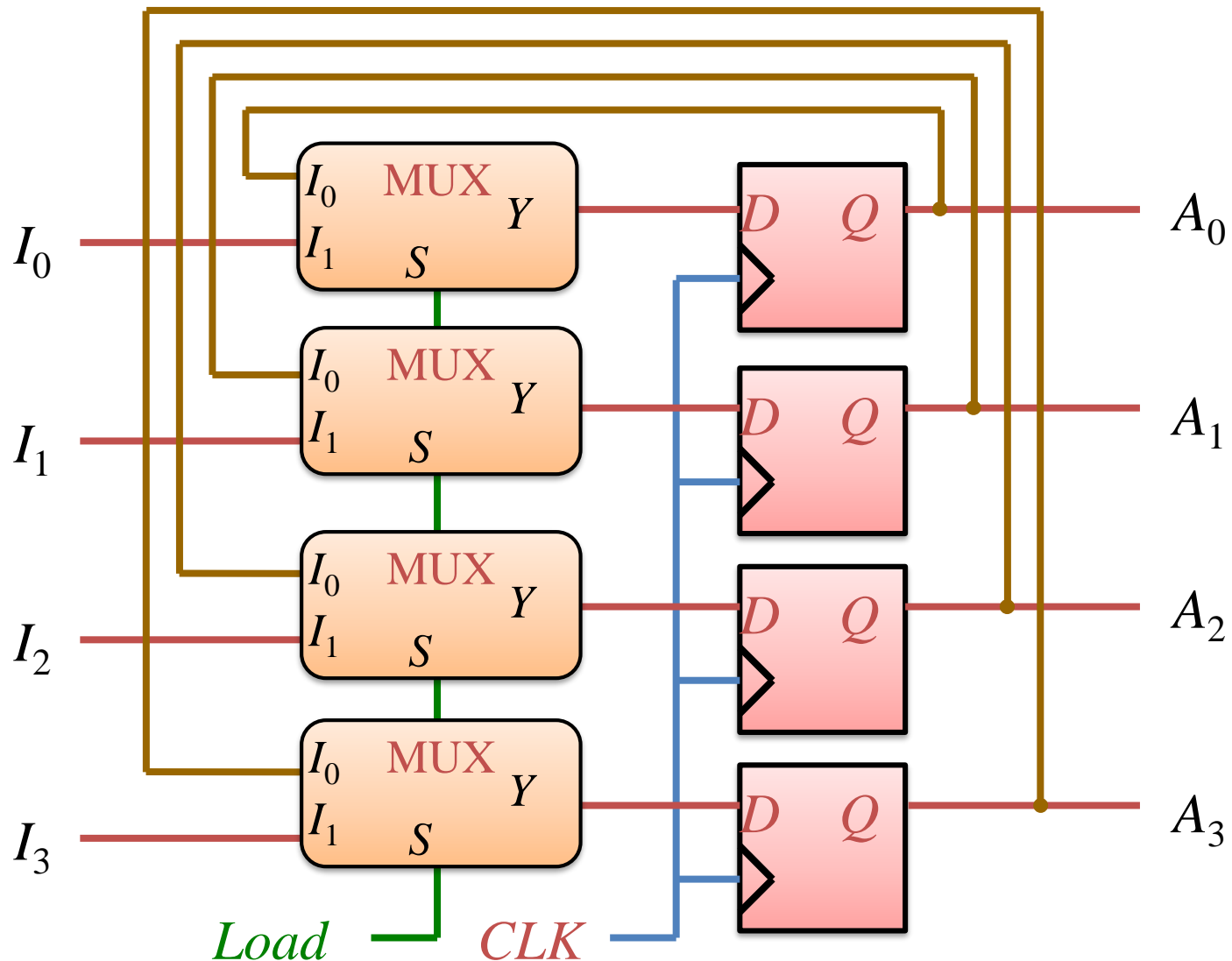
Registers with Parallel Load

- Should we block the “Clock” to keep the “Data”?



Registers with Parallel Load

- Circulate the “old data”



Shift Registers

- Register (Set of FFs)
- 4-Bit Shift Register (Example)
- Serial in Serial Out (SISO)

Shift Register

- Right Shift Example (Left shift is similar)
 - Move each bit one position right
 - Shift in 0 to leftmost bit

Q: Do four right shifts on 1001, showing value after each shift

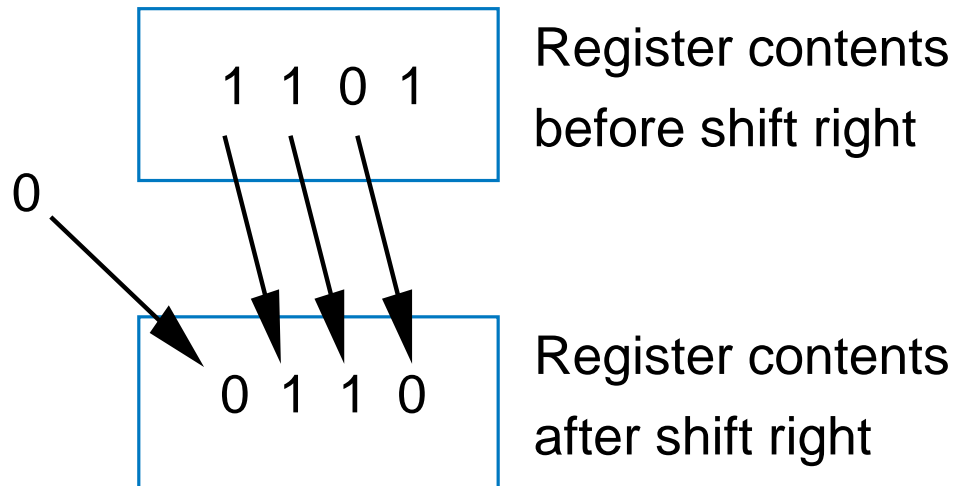
A: 1001 (original)

0100

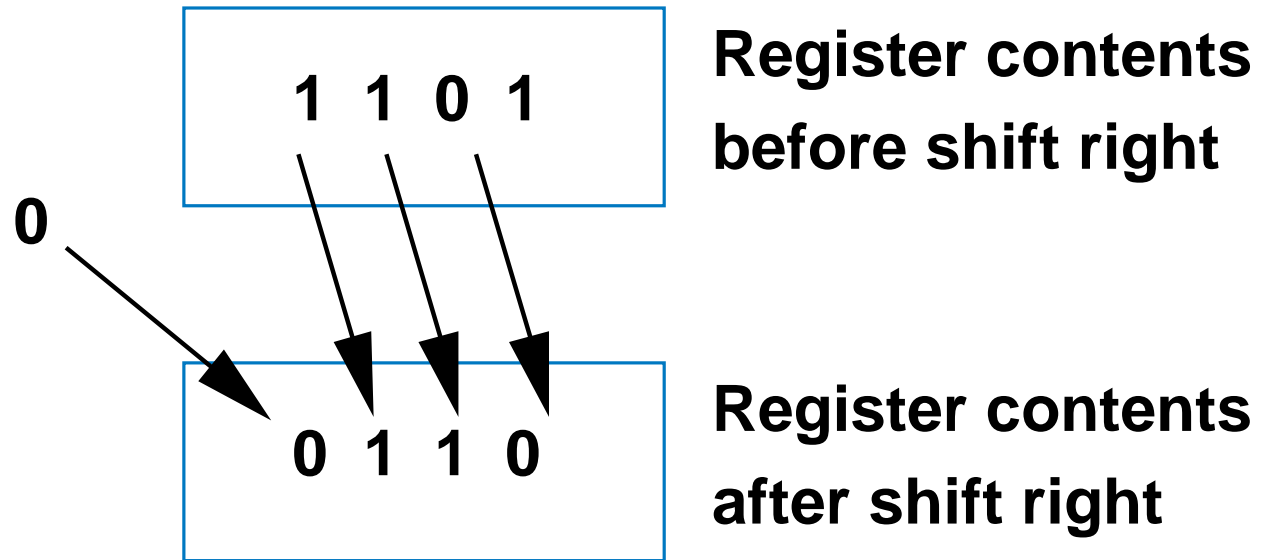
0010

0001

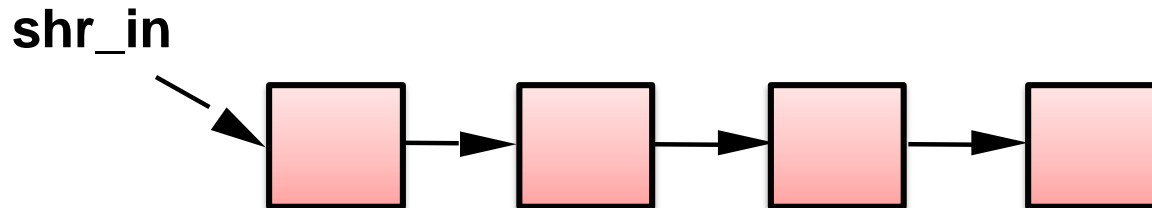
0000



Shift Register

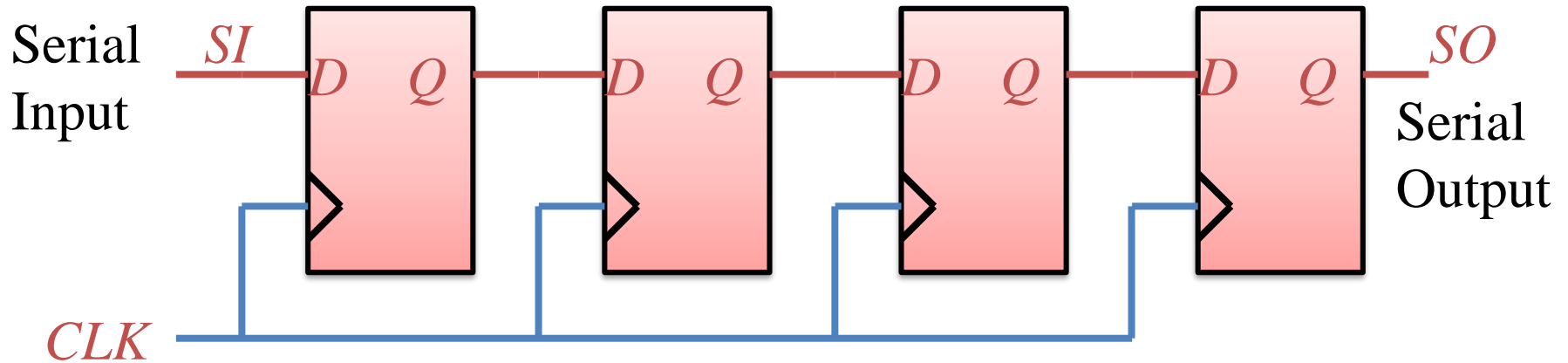


- **Implementation: Connect flip-flop output to next flip-flop's input**

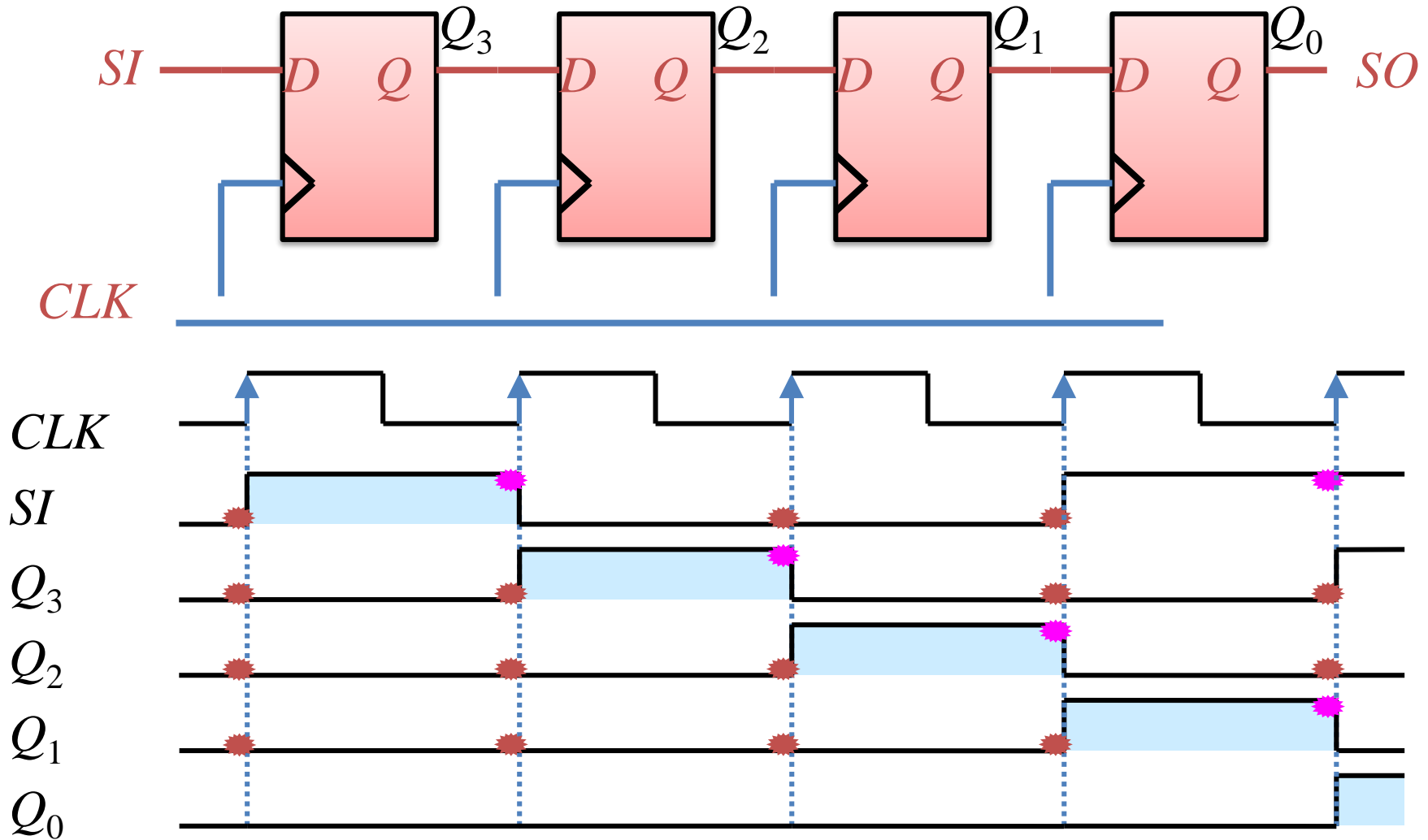


Shift Registers

- 4-Bit Shift Register
- Serial in Serial Out (SISO)

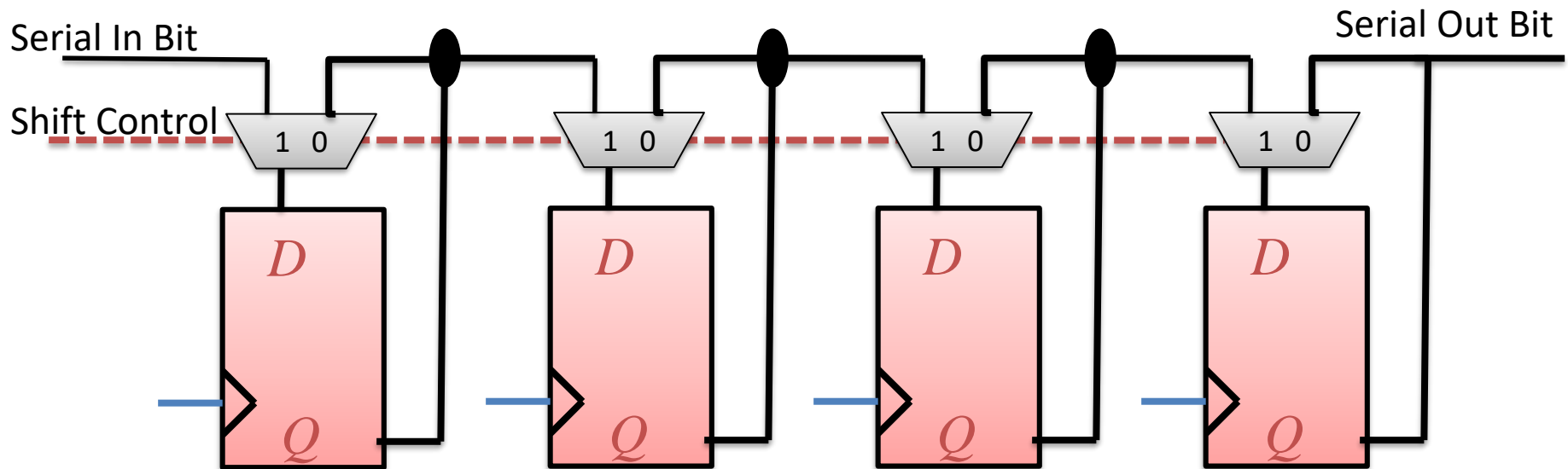


Shift Registers (SISO)



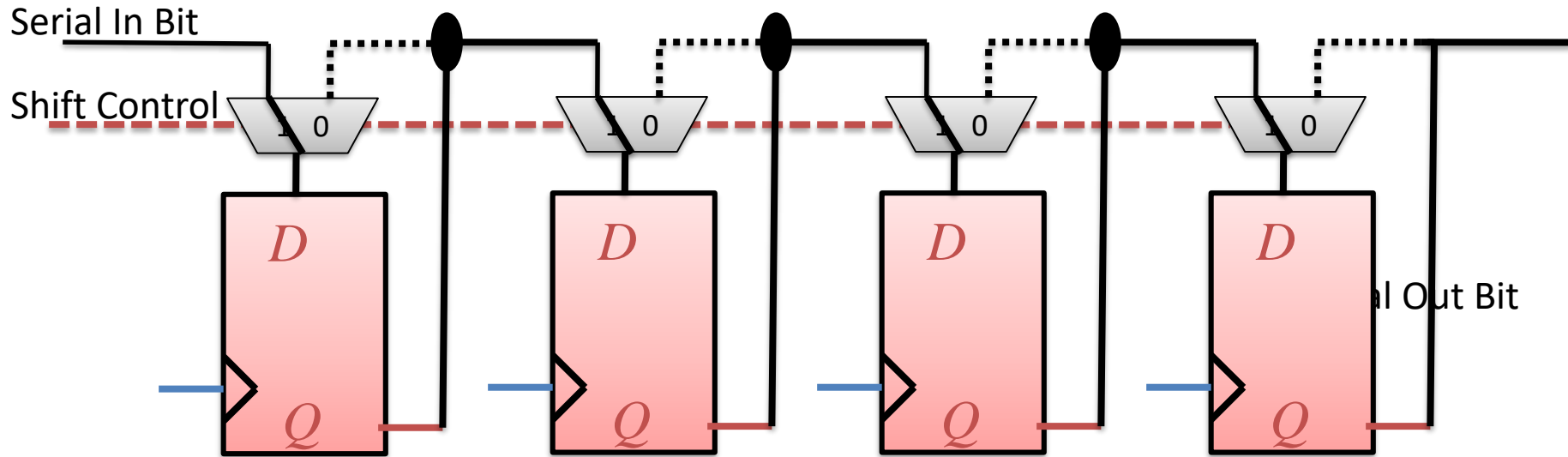
Shift Register with Control

- To allow register to either shift or retain, use 2x1 muxes
 - **Shift Control: shr: 0 means retain, 1 shift**
 - shr_in: value to shift in : May be 0/1, or right most bit



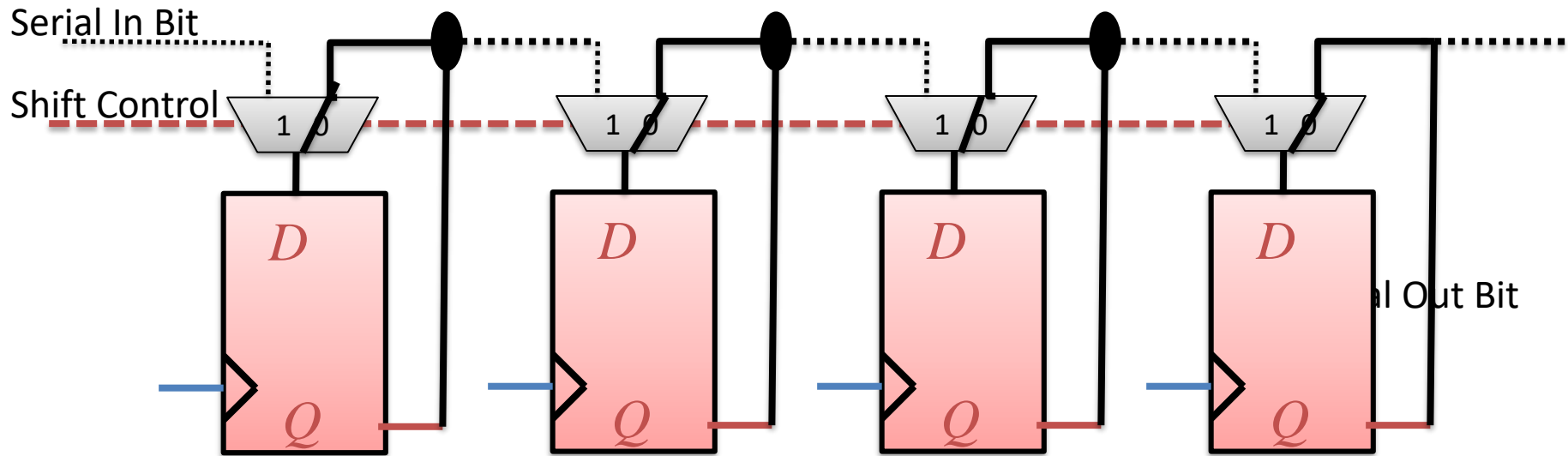
Shift Register with Control

Shift Control=1, Do one right shift per cycle

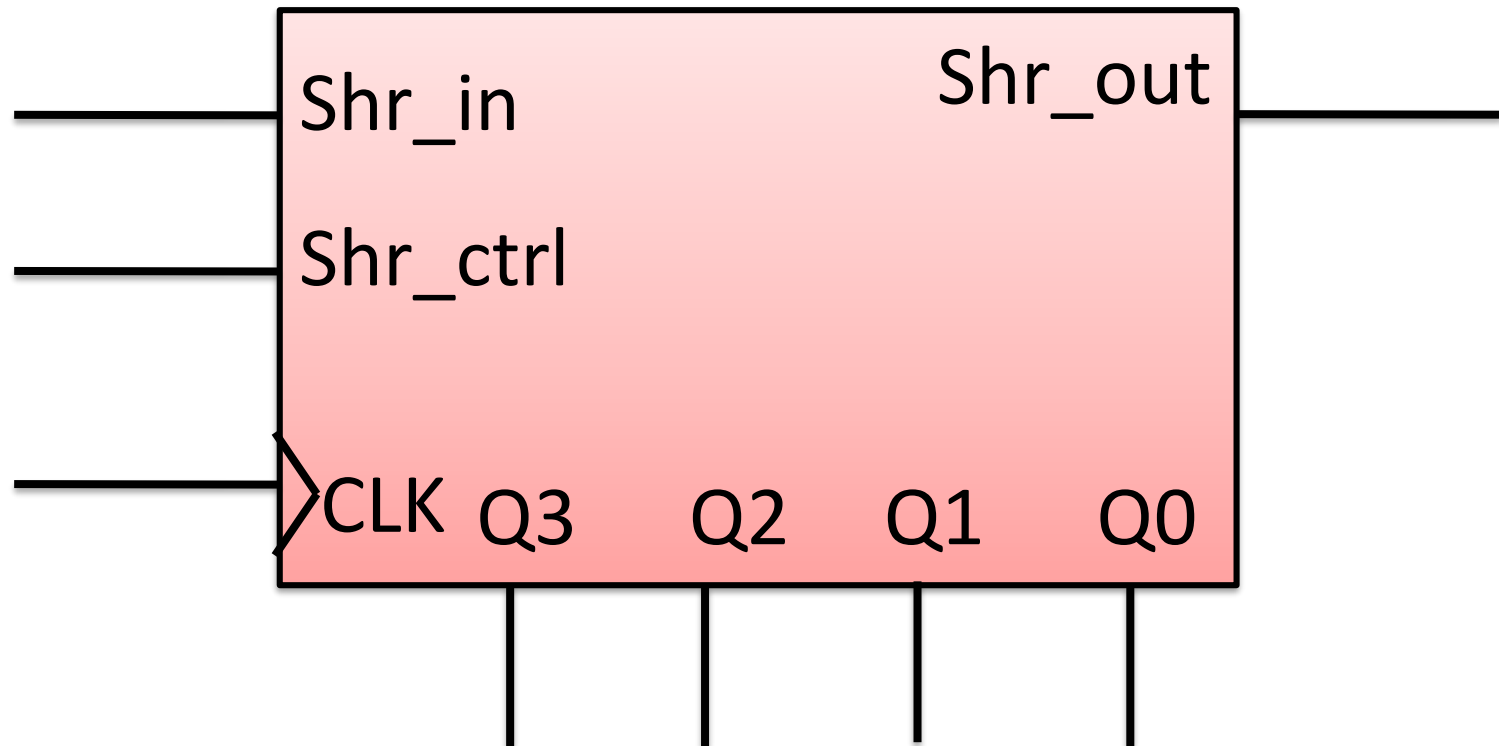


Shift Register with Control

Shift Control=0, No change to Data

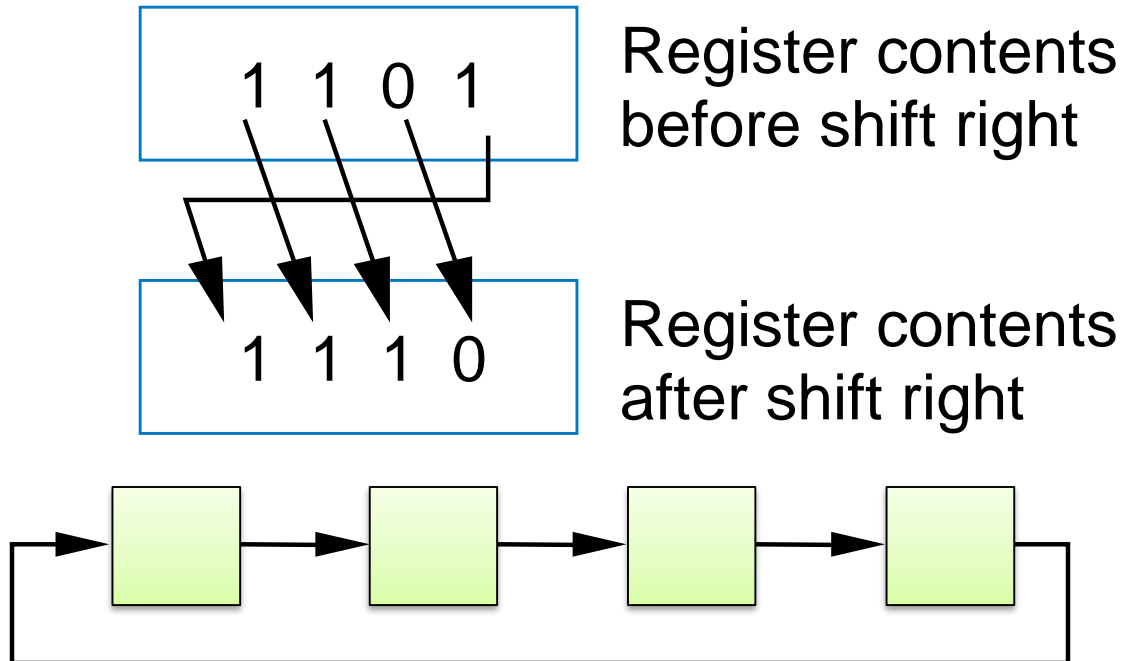


Shift Register with Control

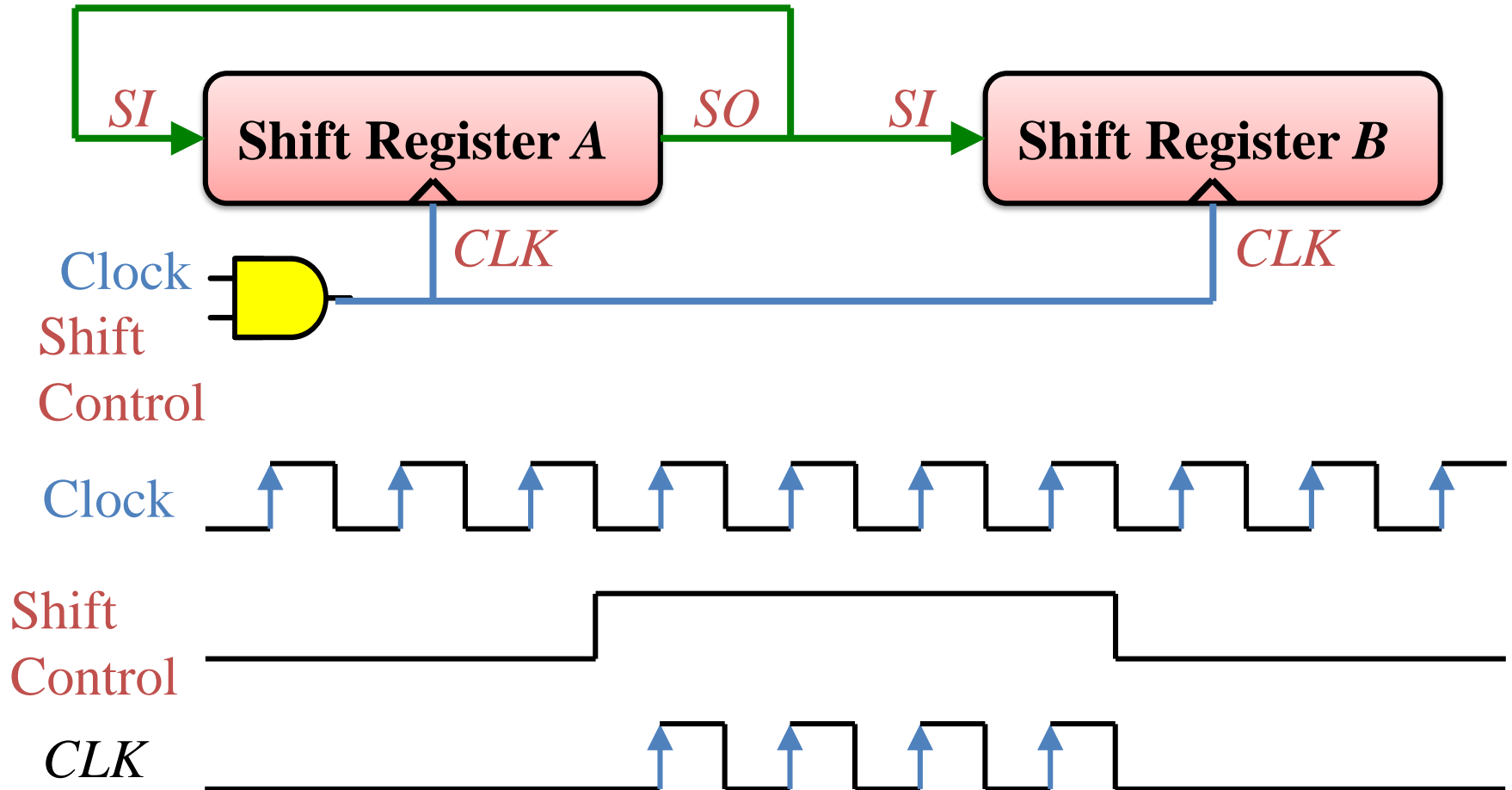


Rotate Register

- Rotate right: Like shift right, but leftmost bit comes from rightmost bit

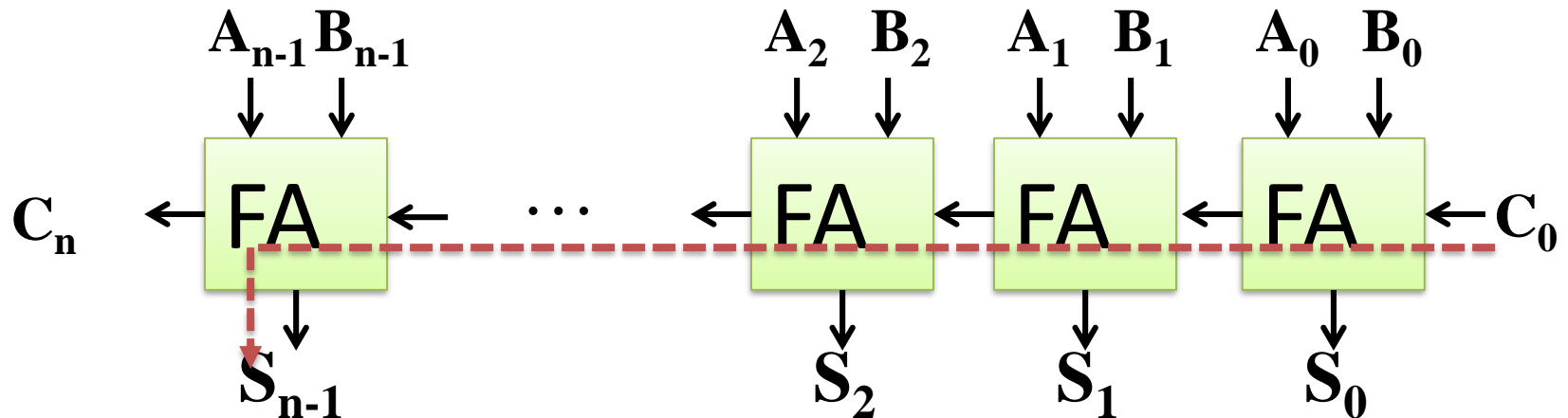


Serial Transfer

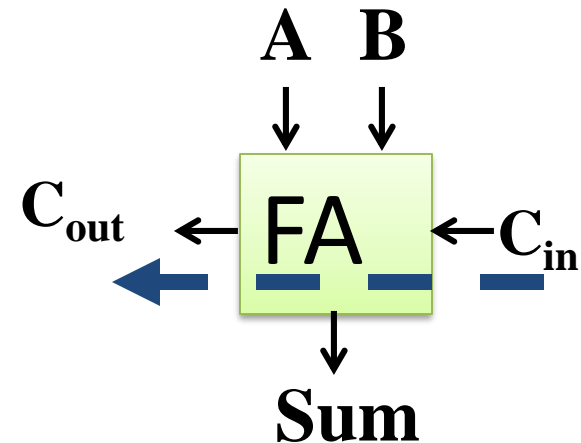


N-Bit RCA: Series of FA Cells

- To add two n-bit numbers

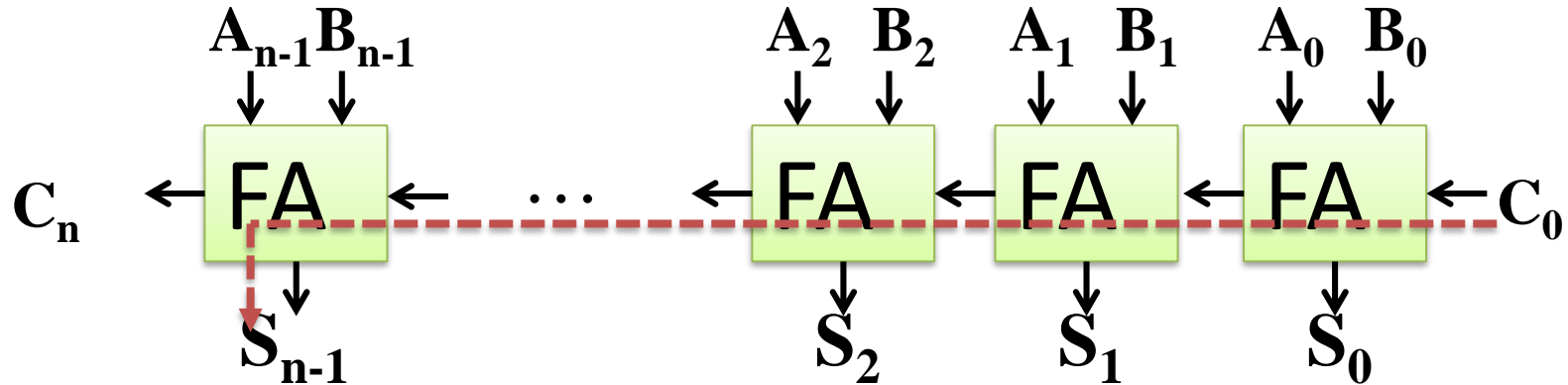


- Adder delay = $T_c * n$
- $T_c = (C_{in} \text{ to } C_{out} \text{ delay}) \text{ of a FA}$



N-Bit Ripple-Carry Adder

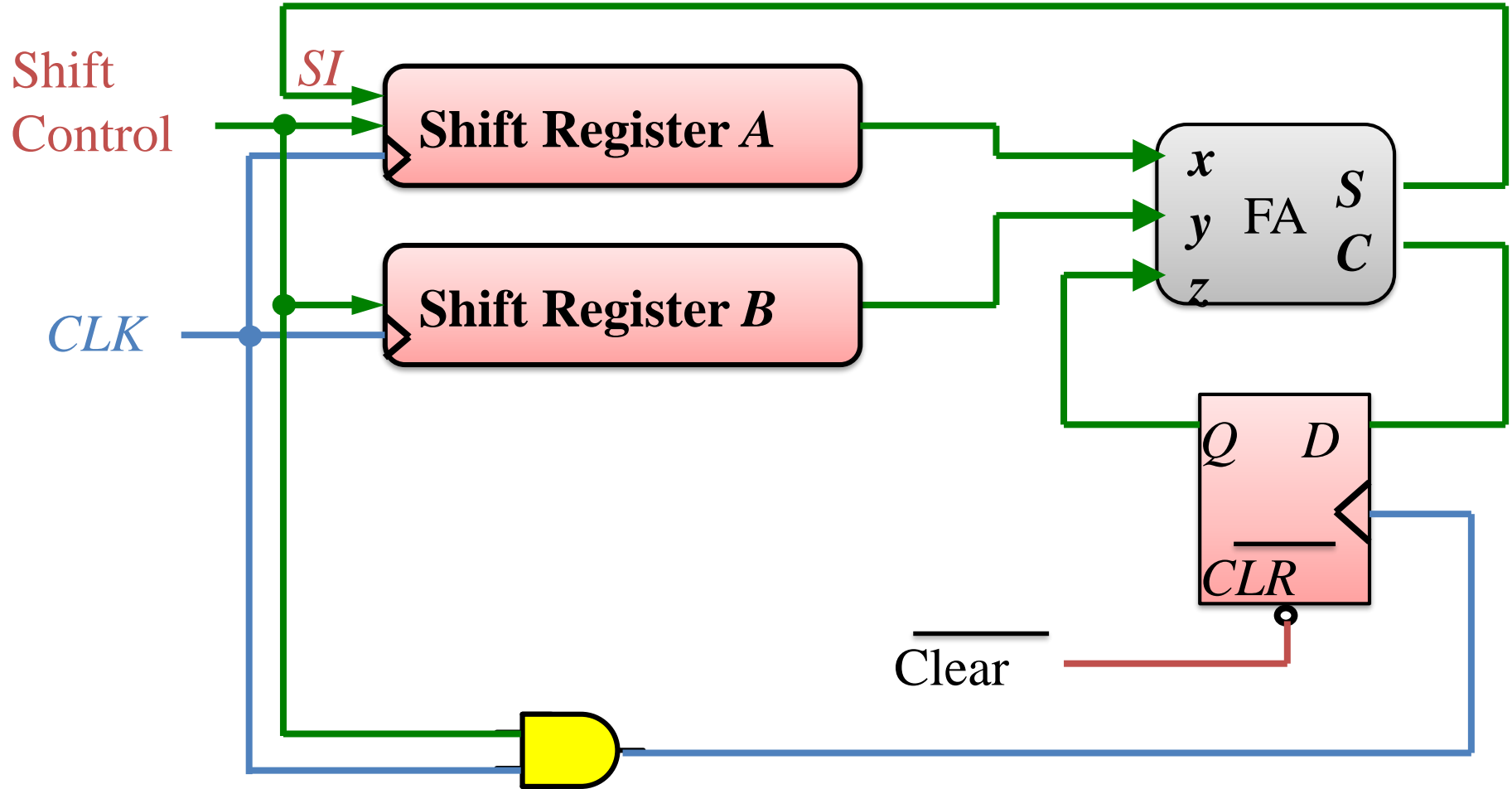
- To add two n-bit numbers



- Any point of time: only one FA is active

Do we require N Full Adder?

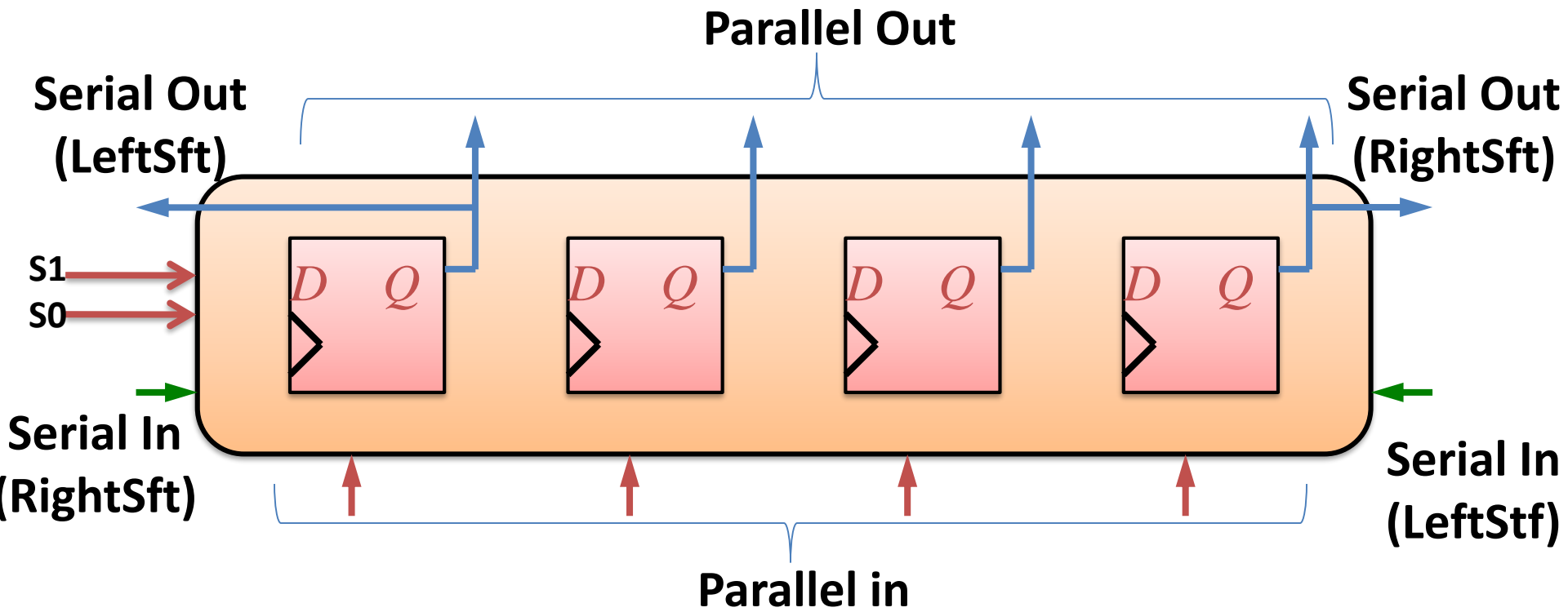
Serial Addition



Universal Shift Register

- Parallel-in Parallel-out (PIPO)
- Serial-in Serial-out (SISO)
- Serial-in Parallel-out (SIPO)
- Parallel-in Serial-out (PISO)

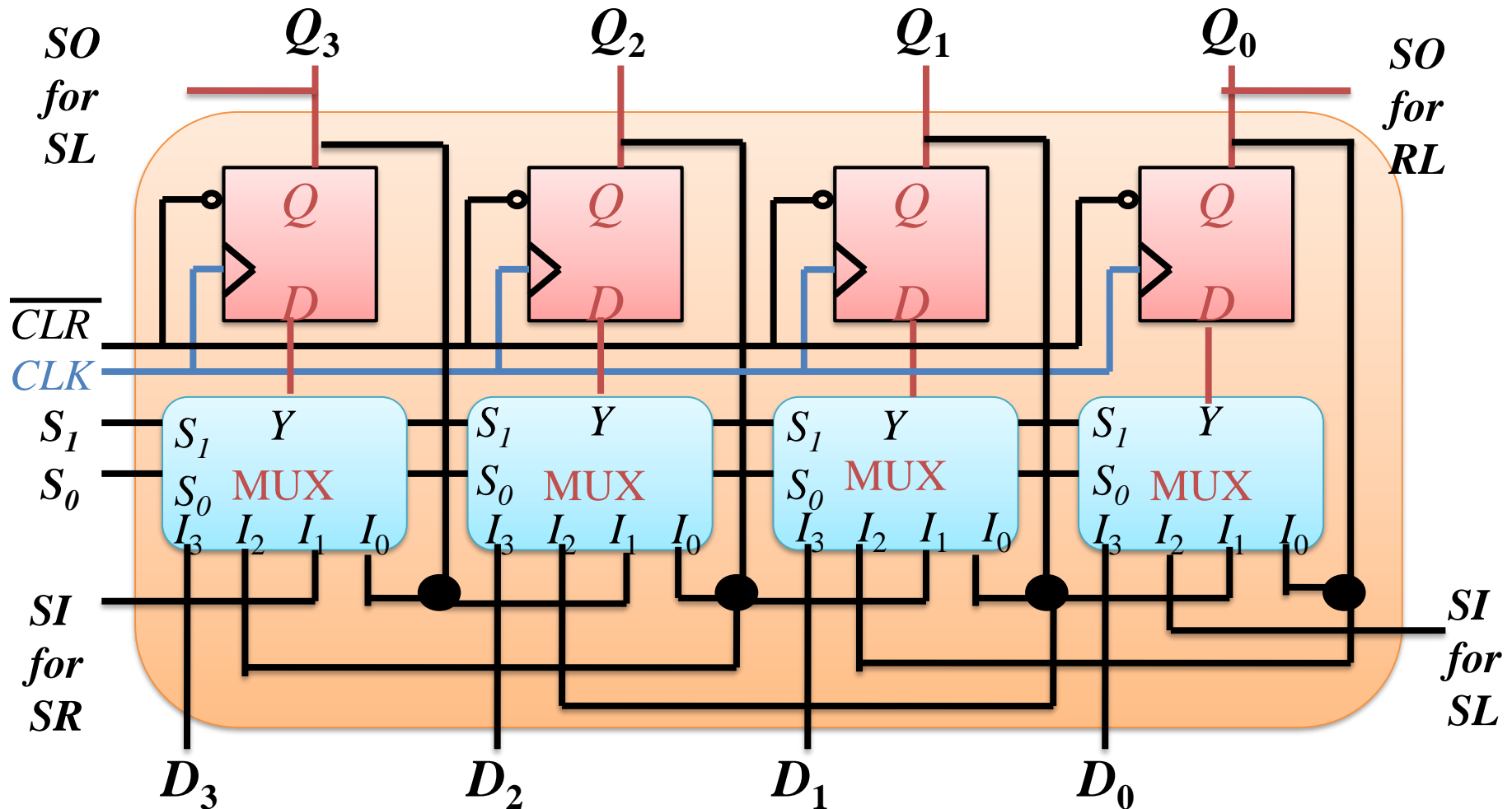
Universal Shift Register



Universal Shift Register

How to design
such an USR?

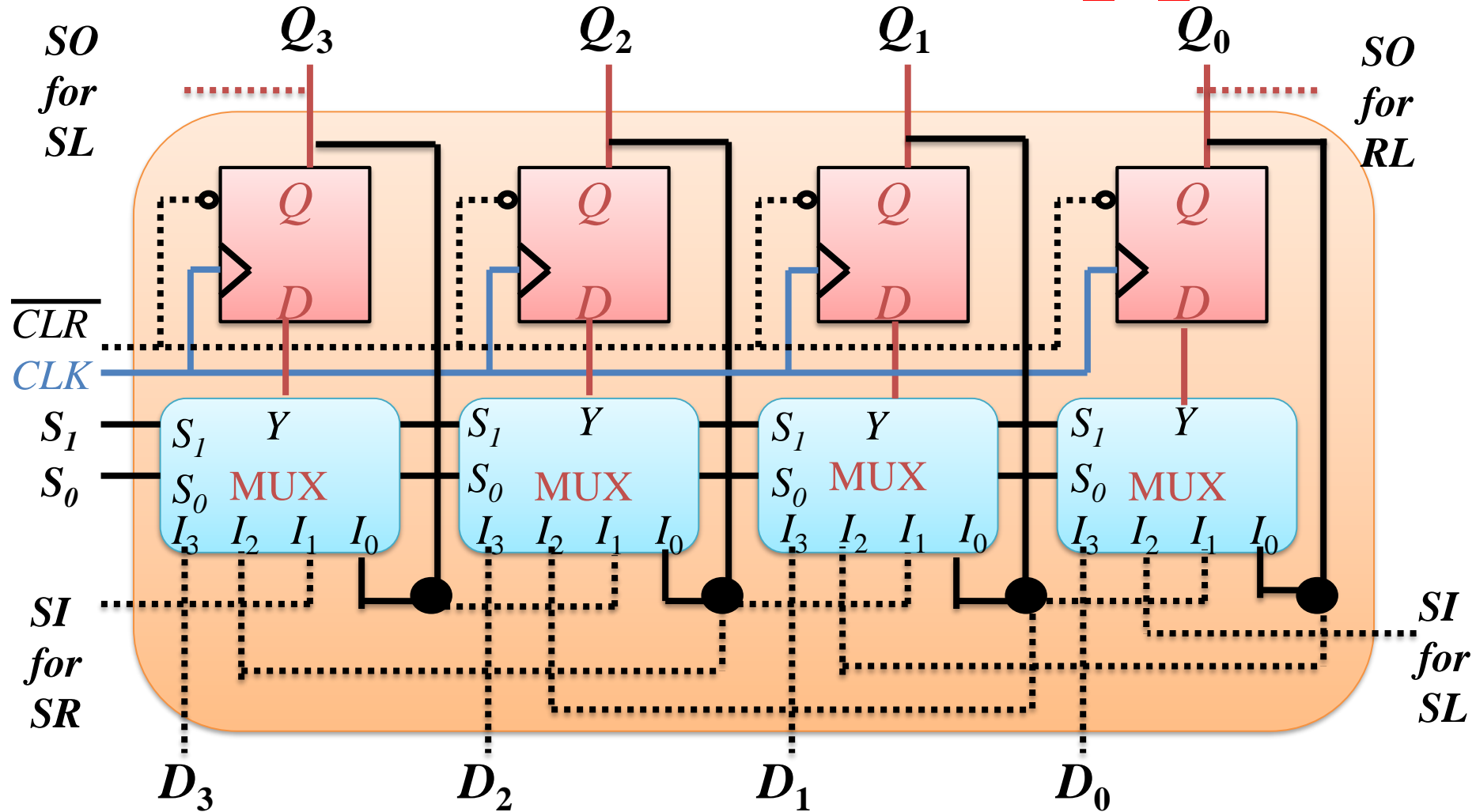
Universal Shift Register



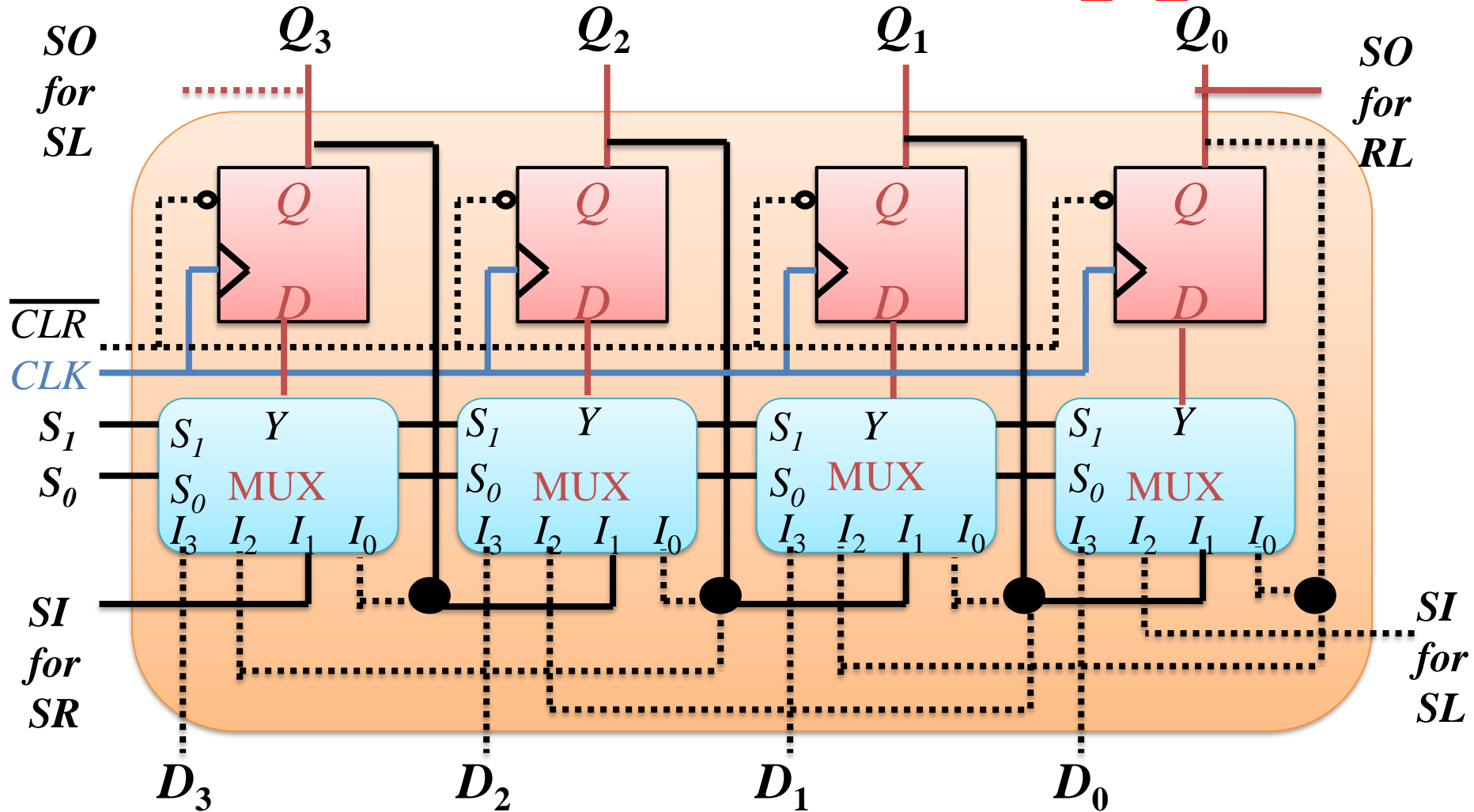
Operation Table for USR

S1	S0	Register Operation
0	0	No Change
0	1	Shift Right
1	0	Shift Left
1	1	Parallel Load

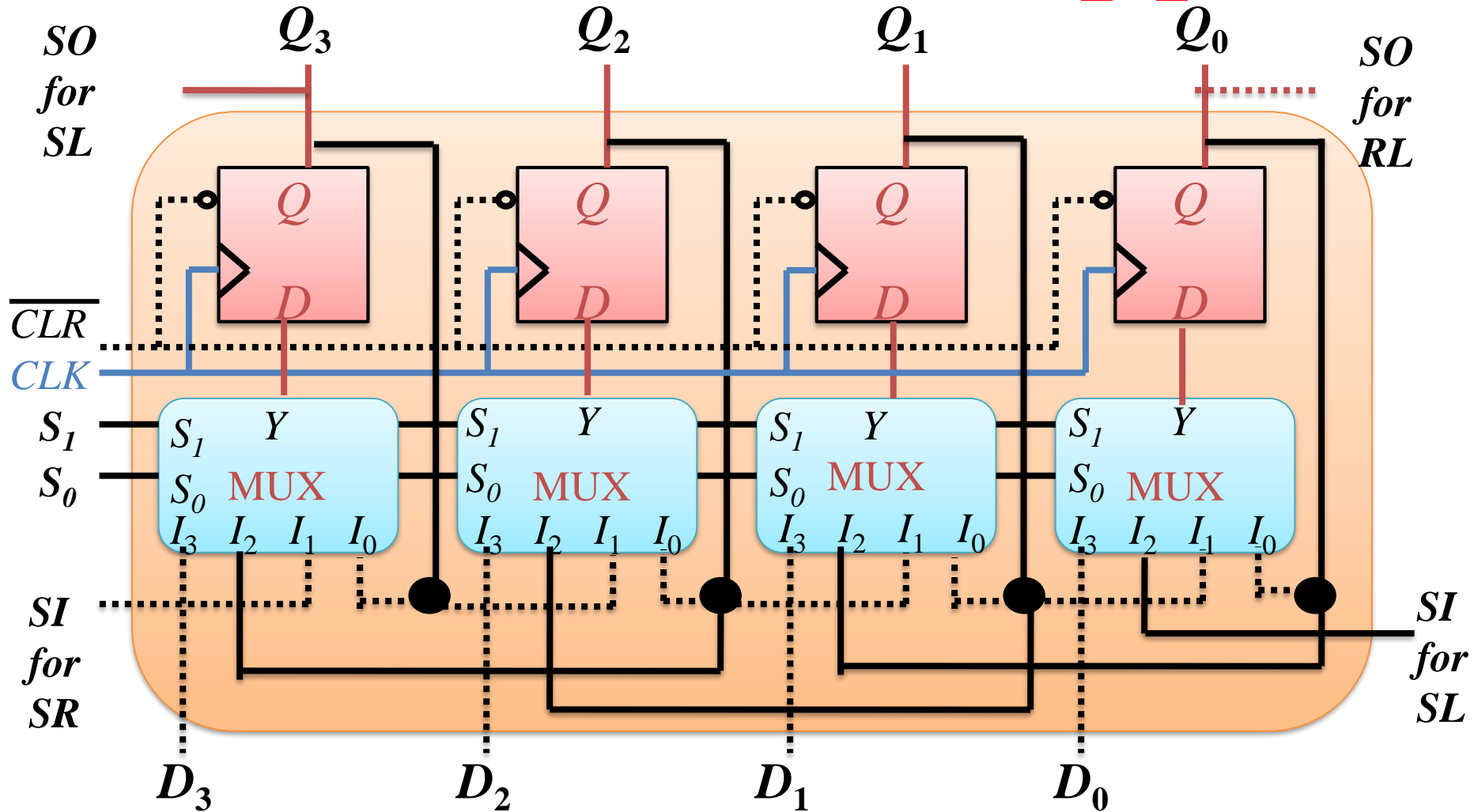
Universal Shift Register: $S_1S_0=00$



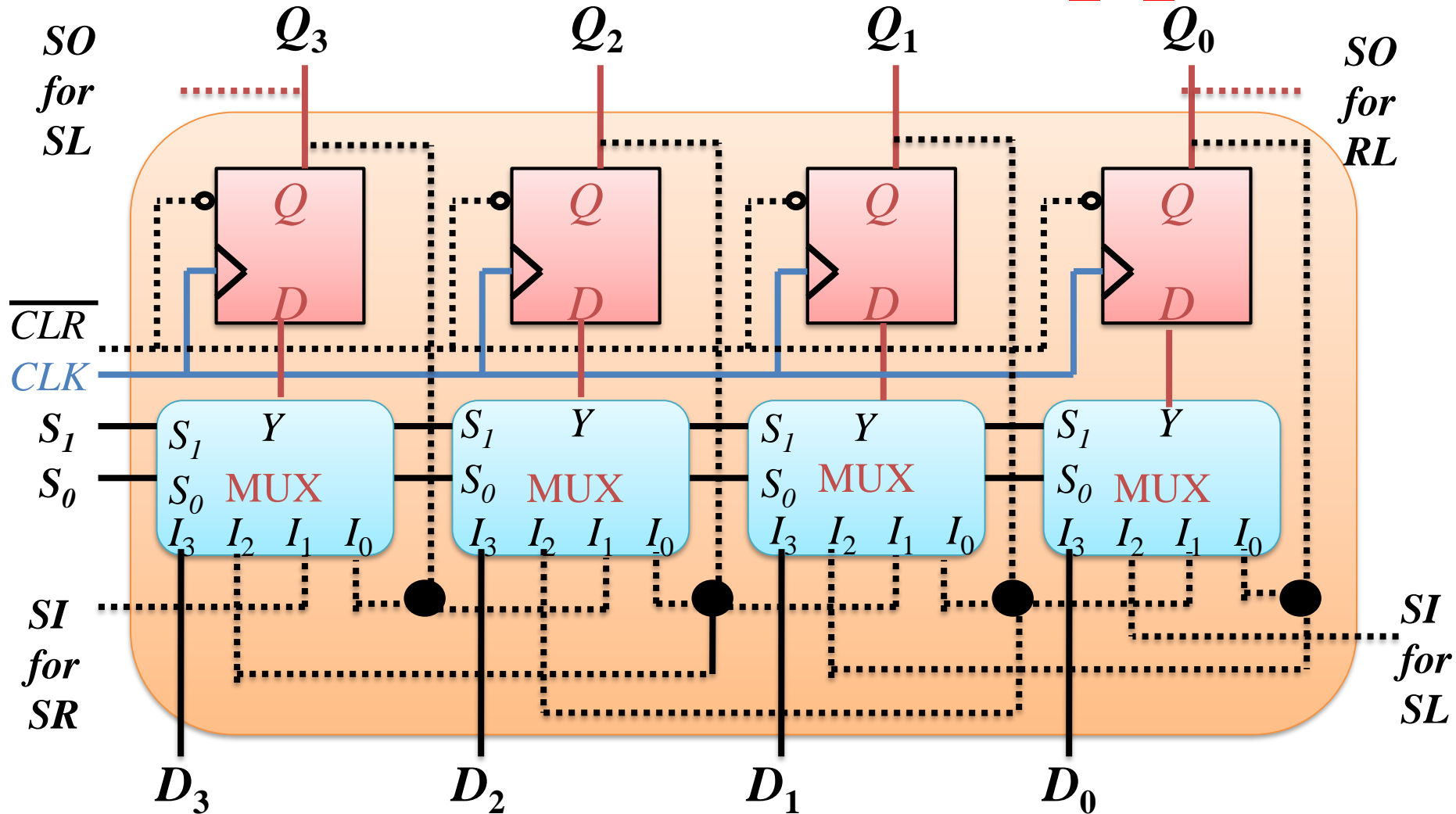
Universal Shift Register: $S_1S_0=01$



Universal Shift Register: $S_1S_0=10$



Universal Shift Register: $S_1S_0=11$



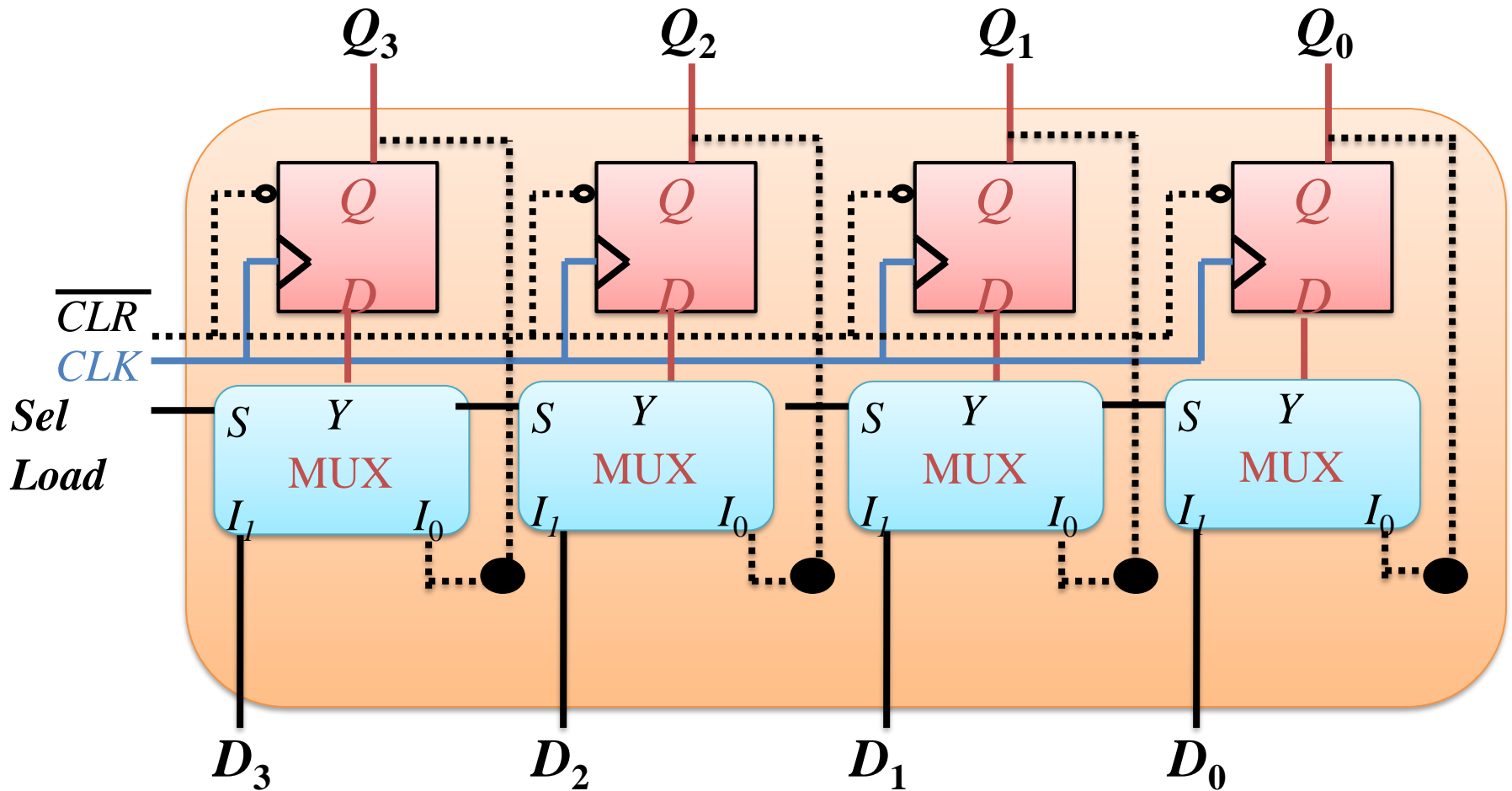
Design of Multifunction Register

- M function N bit Register
 - Last example : 4 functions and 4 bits register
- N number of $\text{ceil}[\text{Log}_2 M] \times 2^{\text{ceil}[\text{Log}_2 M]}$ size Multiplexor (Mux)
 - For 4 function 2x4 Mux (**4 to 1 Mux**), 2 select line
 - For 5 function 3x8 Mux (**8 to 1 Mux**), 3 select line
- Extra functions of Mux need to shunted (ignored) to do no work
 - For 5 function need a 3x8 Mux, we need to shunt $8-5=3$ lines shunted (ignored) to do no work

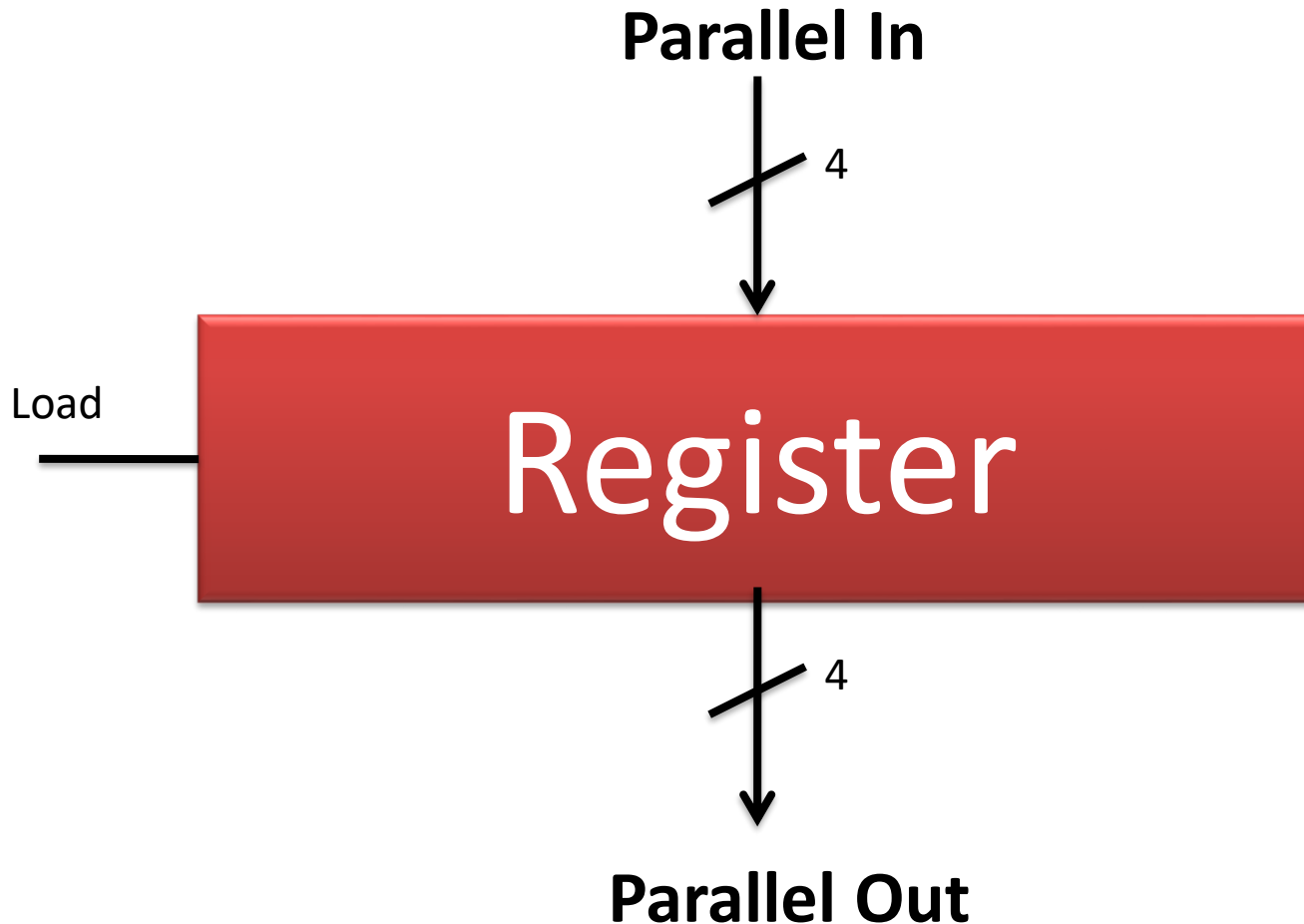
Memory

- Universal register : One register with many functionality
- Many registers clubbed with less functionality
 - No Shift
- Parallel load (In) or Parallel Out to a specific register
- Multiplexors used for selection of registers

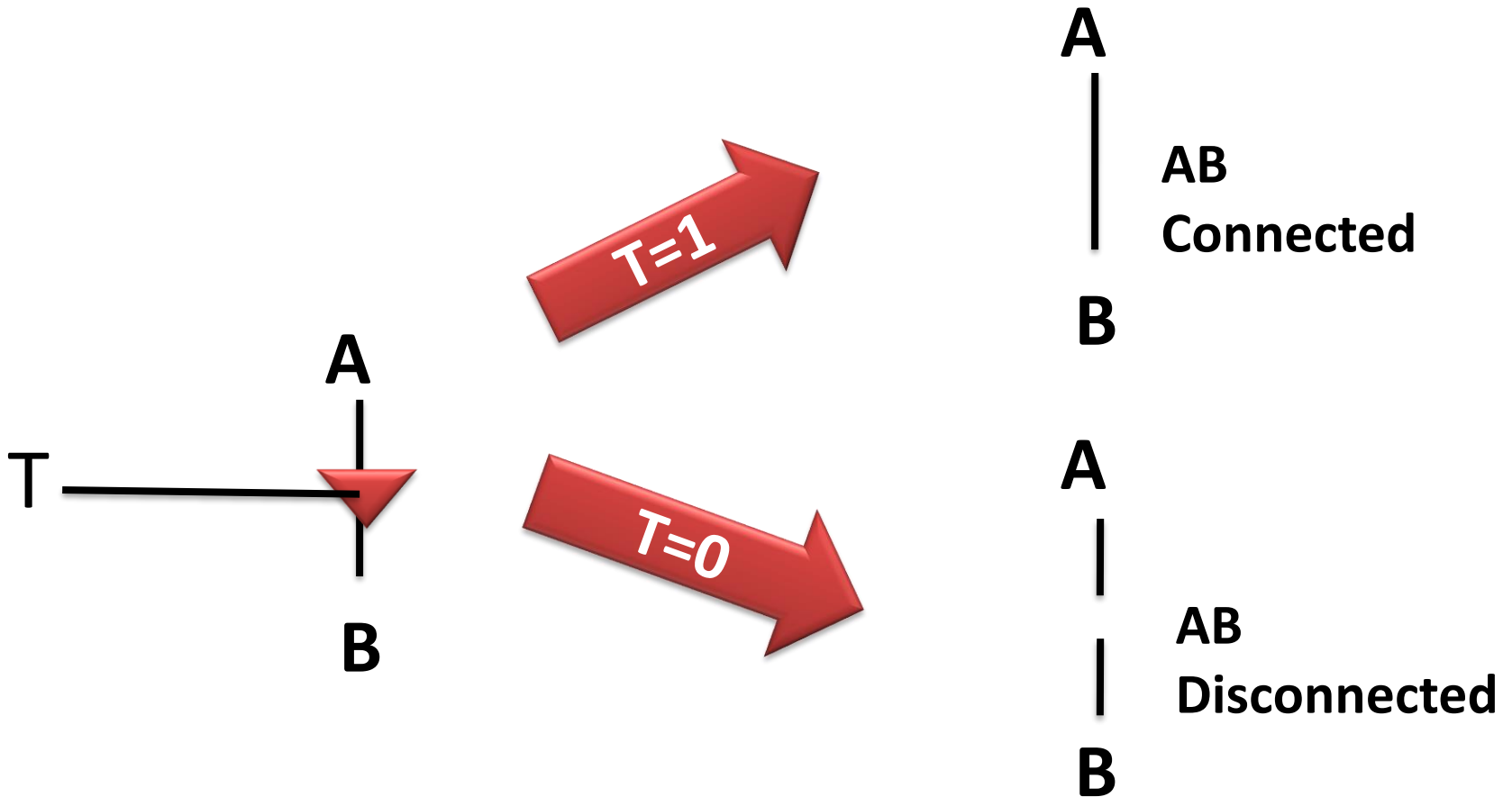
Register(PIPO)



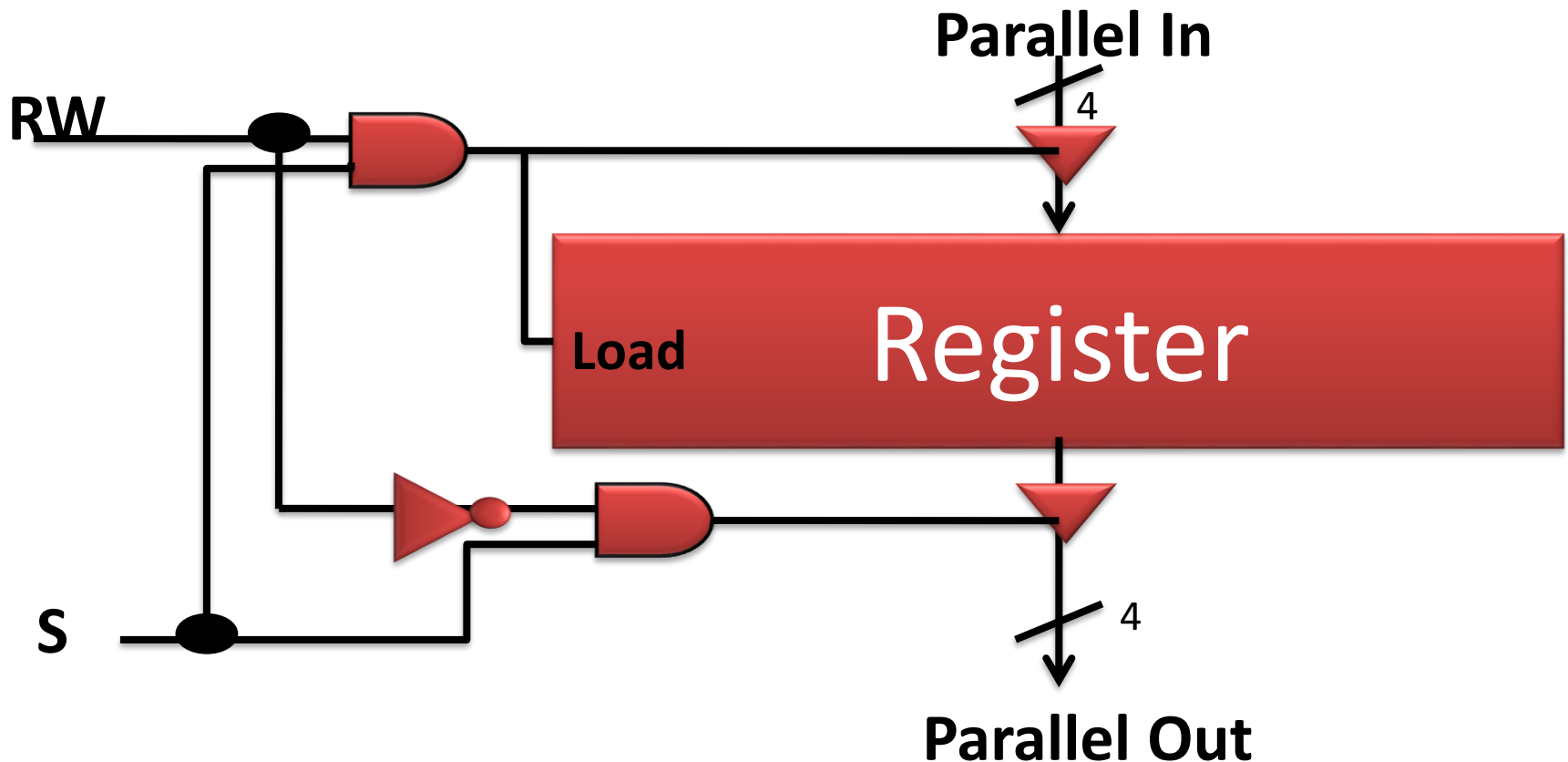
Register(PIPO)



Tri-state Buffer



Register with Read and write control

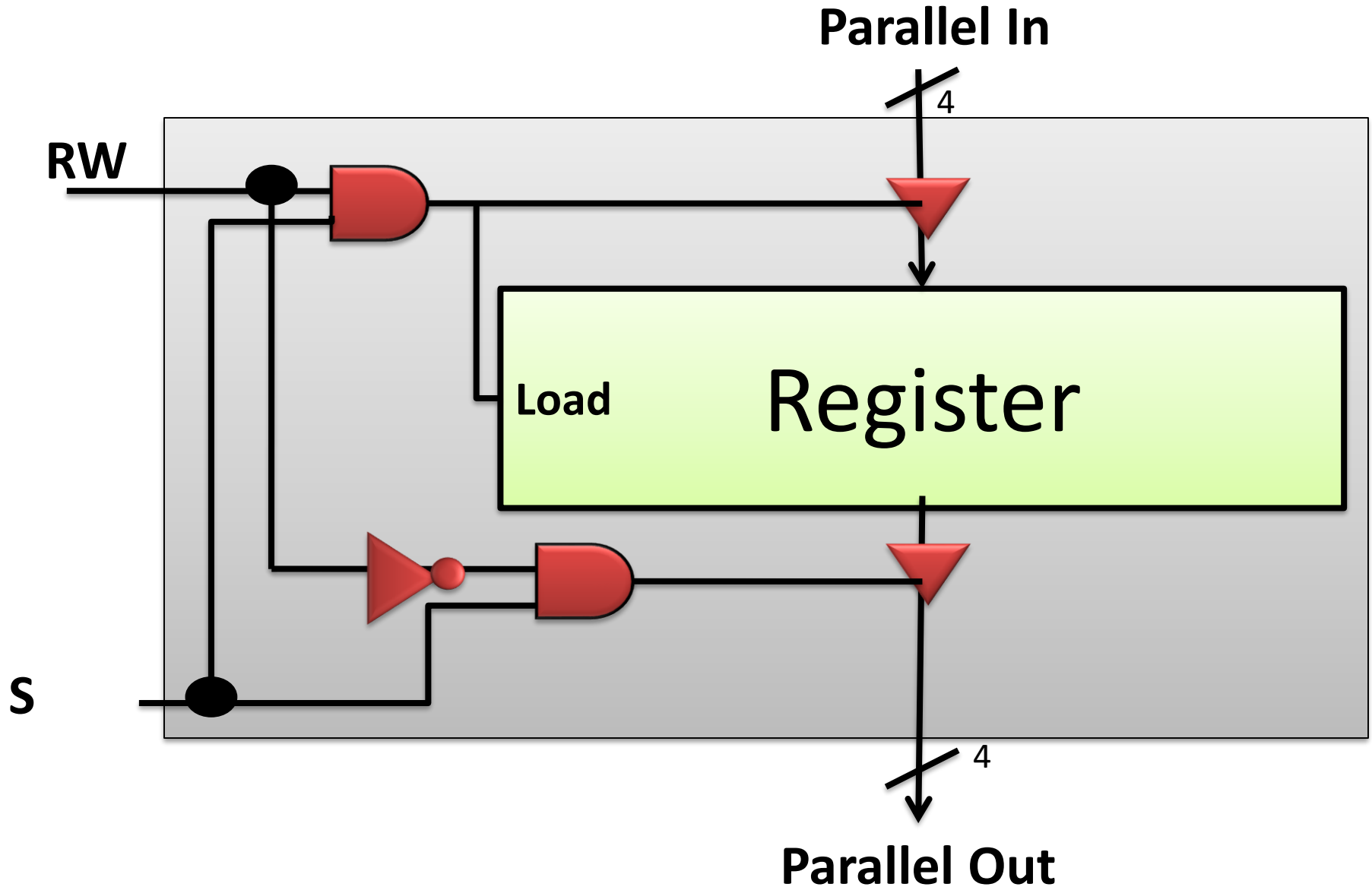


If $S = 0$, TS1 and TS2 will be open

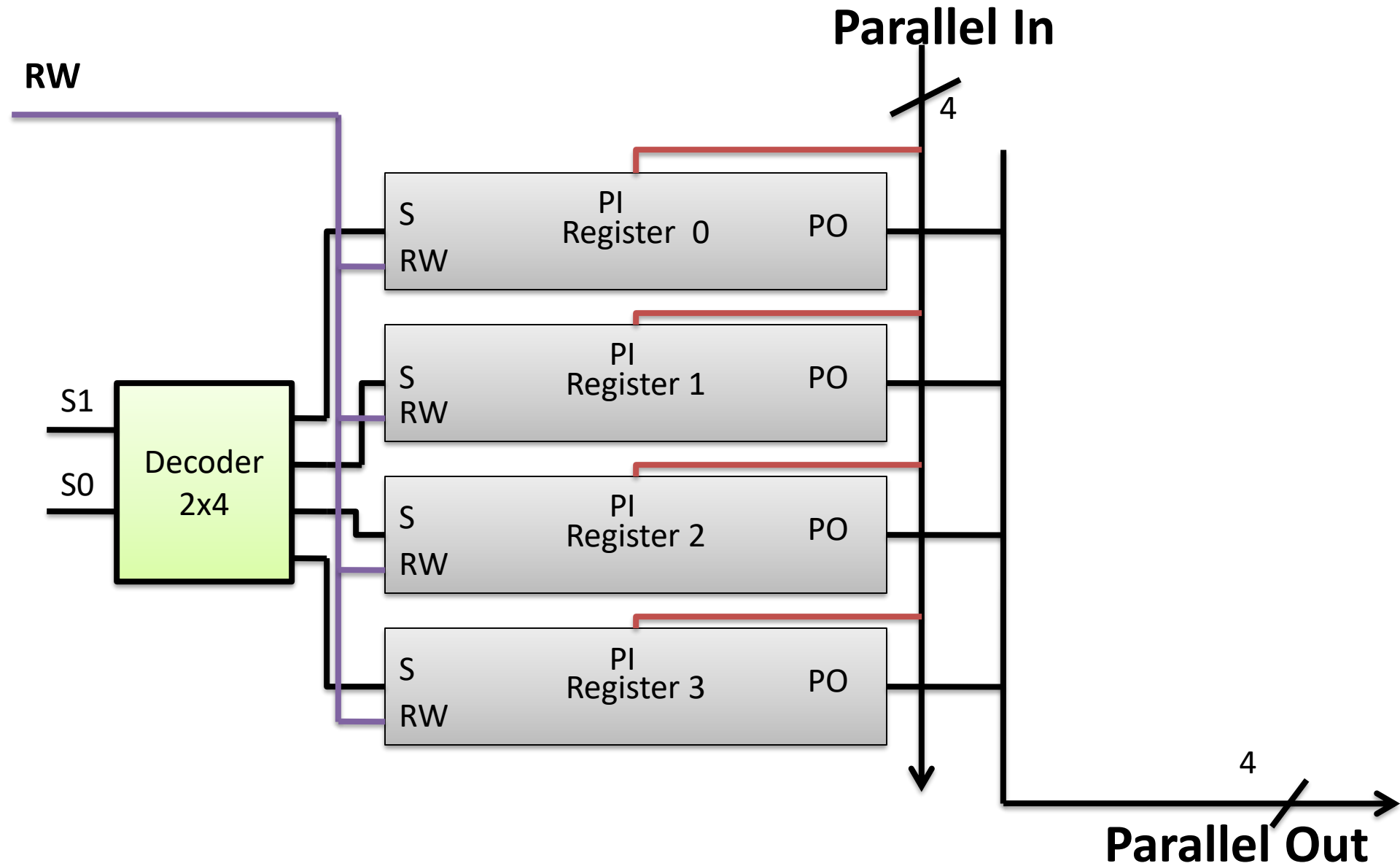
$S=1$, $RW=1$, Write to Register 4 bit of data

$S=1$, $RW=0$, Read data from Register

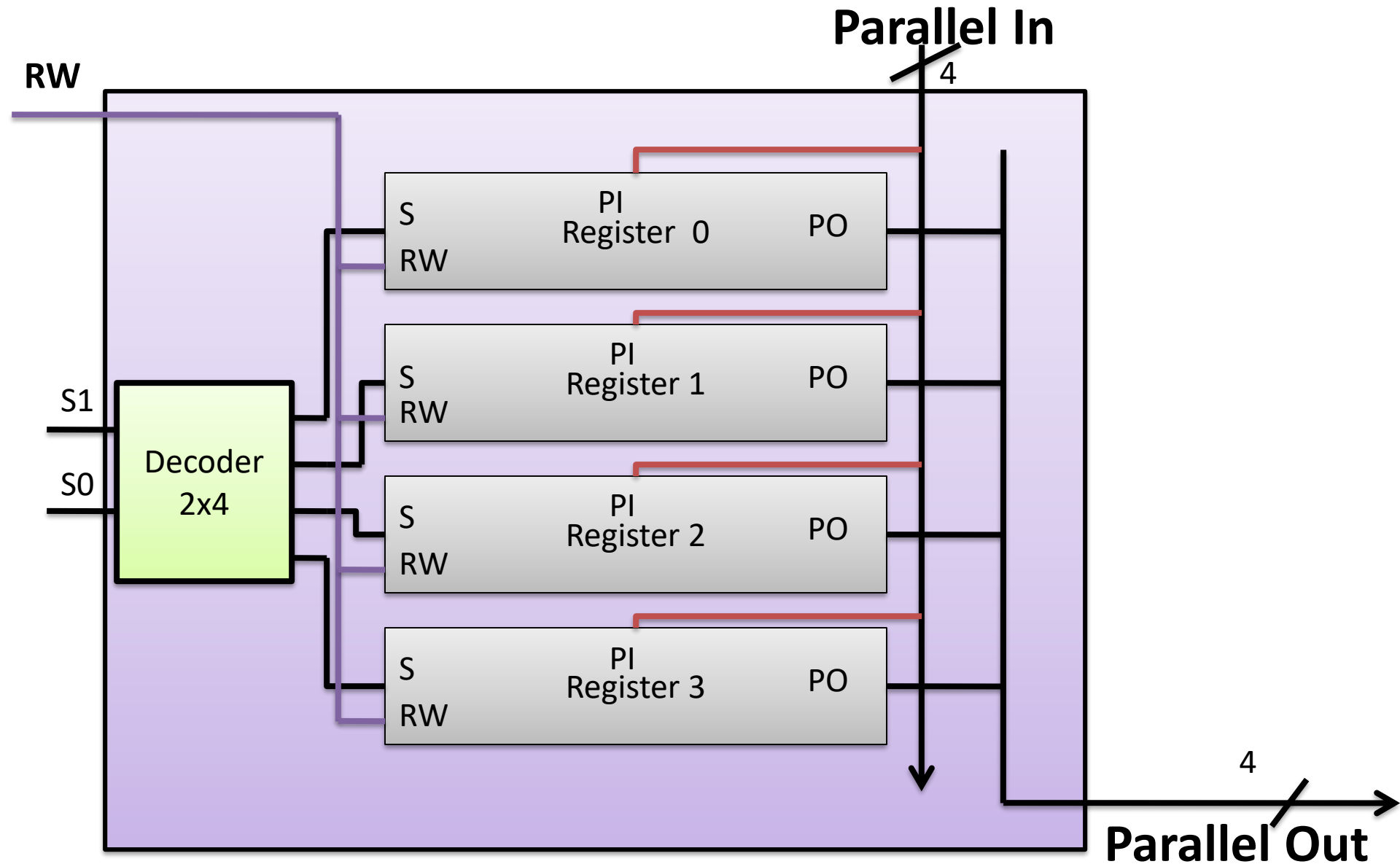
Register with Read and write control



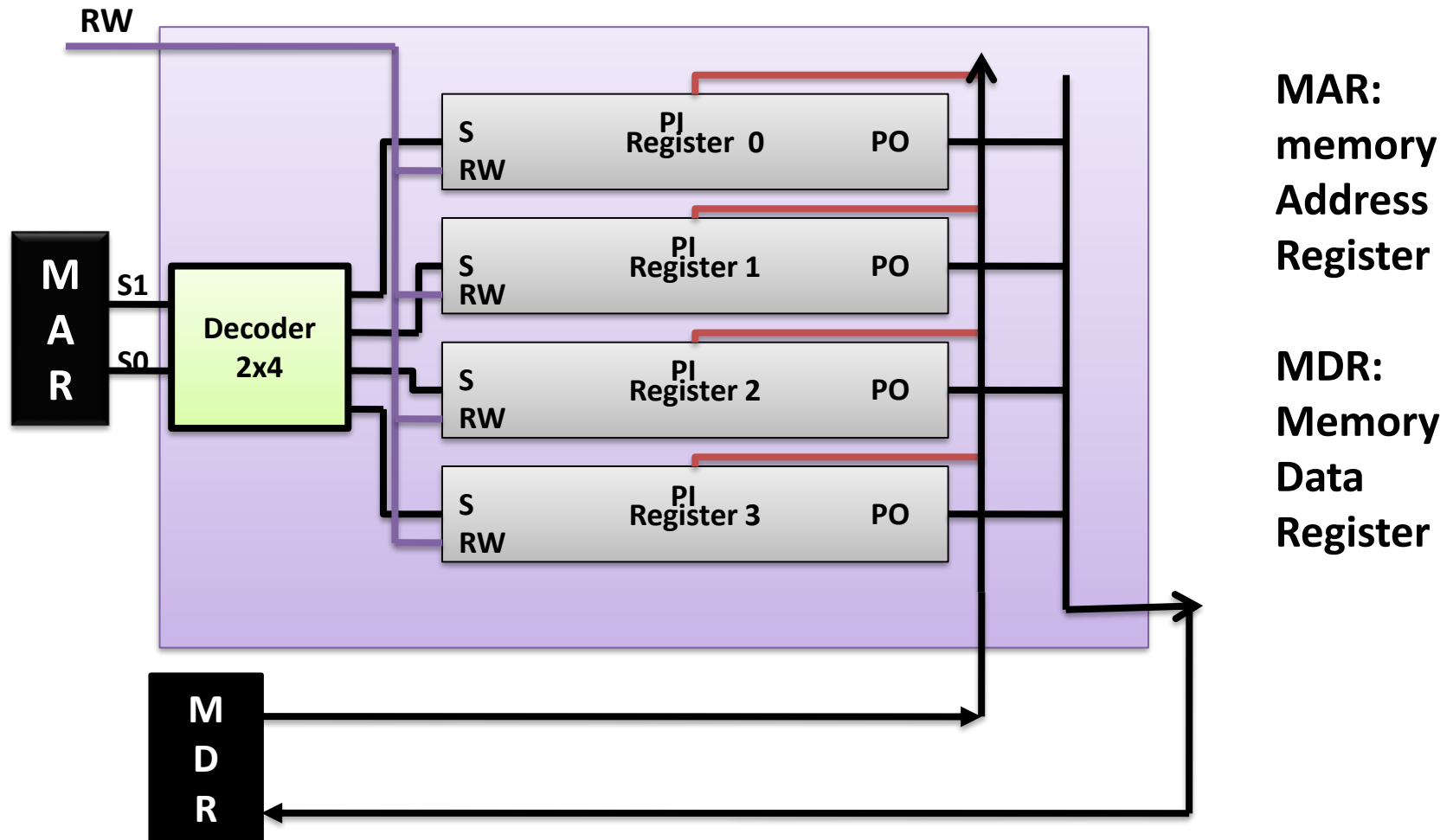
Many Register with RW control



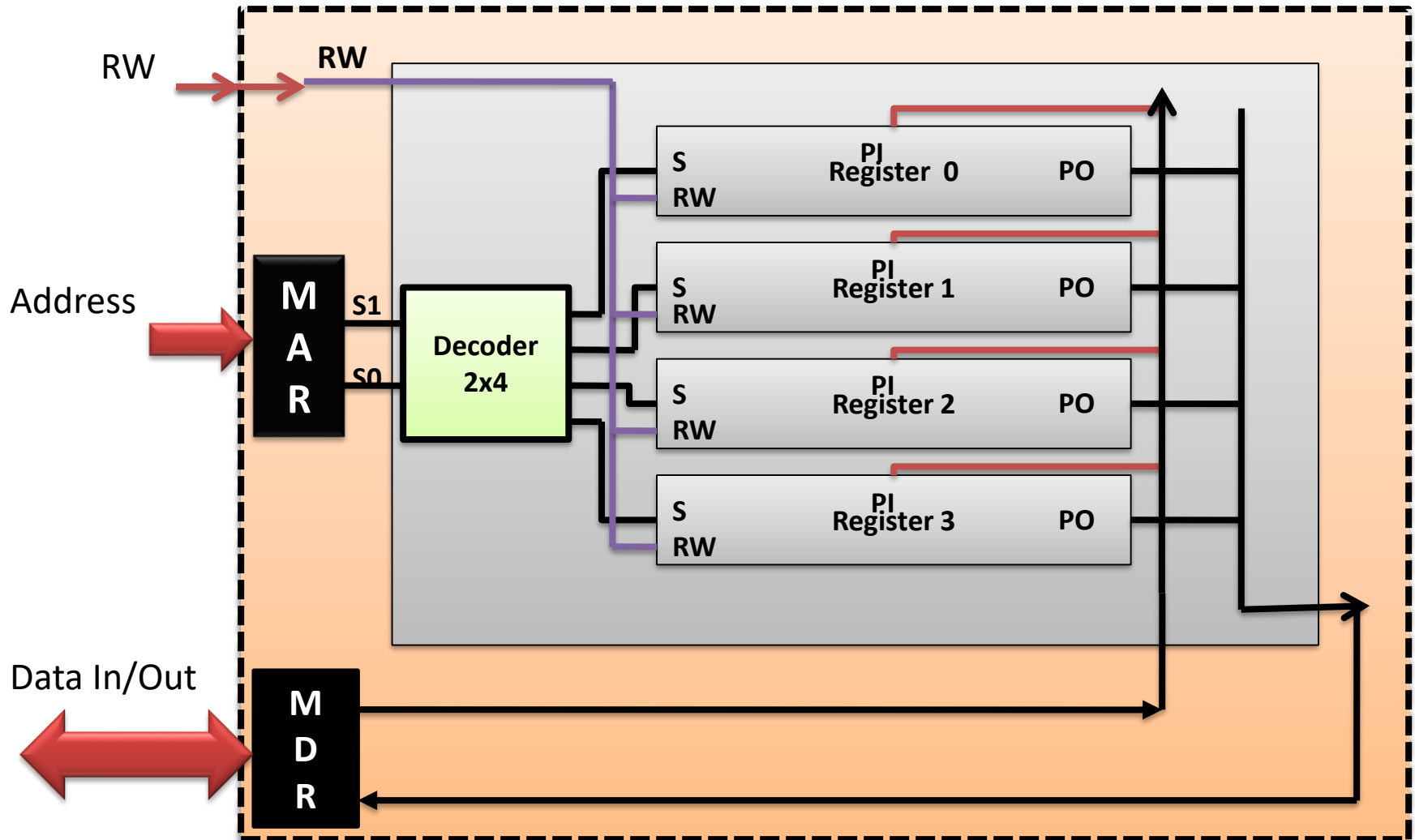
Many Register with RW control



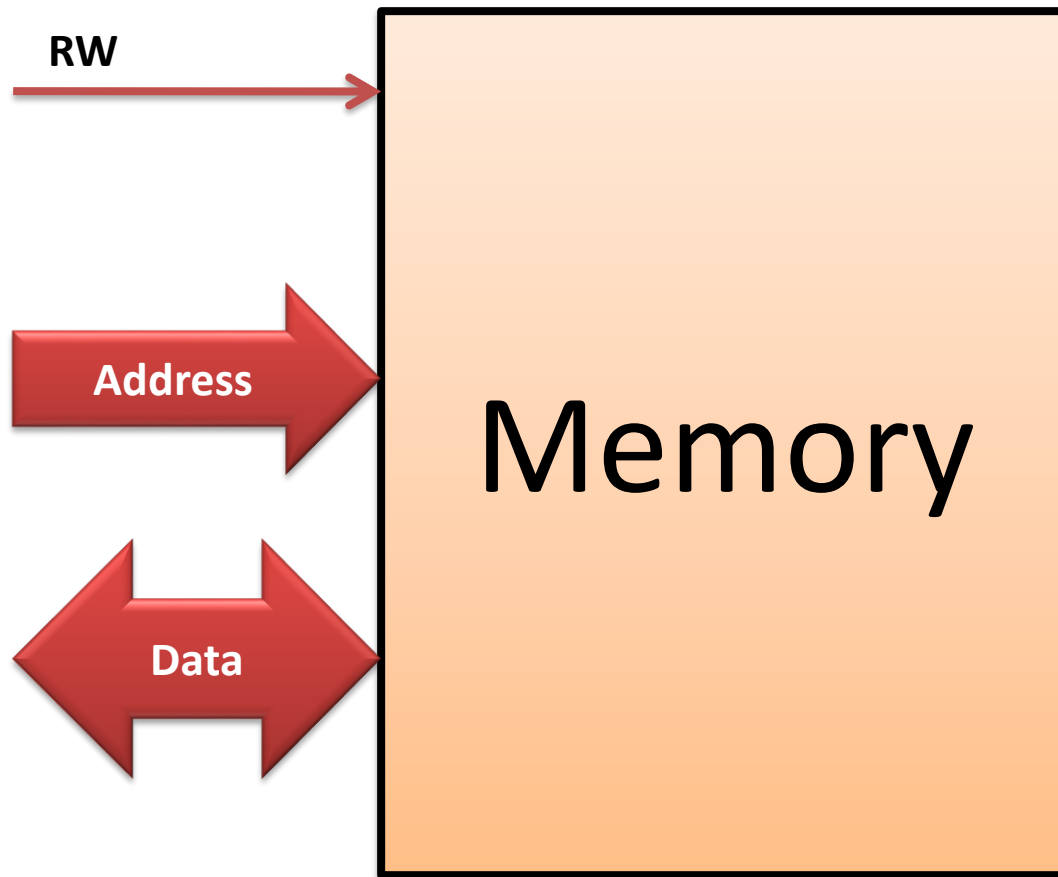
Memory with MAR and MDR



Many Register with RW control: 4 four bit word Memory



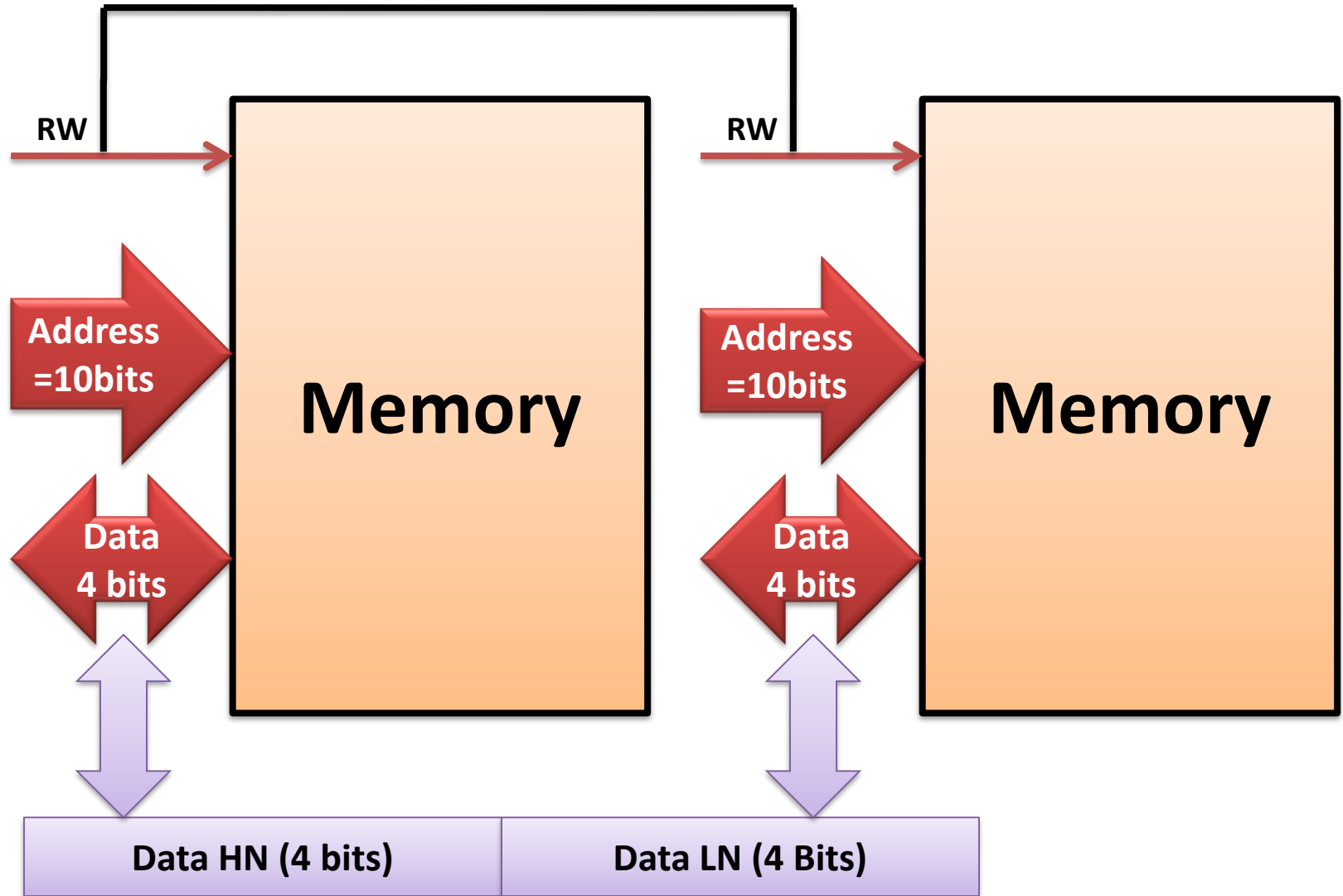
General Memory



Memory : Locations & Word size

- ADDRESS decides the number of memory location
- MDR size : decides the WordSize
- Number of registers : $2 + \text{NumLocations}$
 - $2 + 2^{\text{AddressSize}}$
 - 1 for MBR, 1 for MAR
- $\text{NumFF} = \text{MARsize} + \text{MDRsize} + \text{NumLoc} * \text{MDRSize}$

Designing 1KB Mem using two 1KN Mem



Thanks