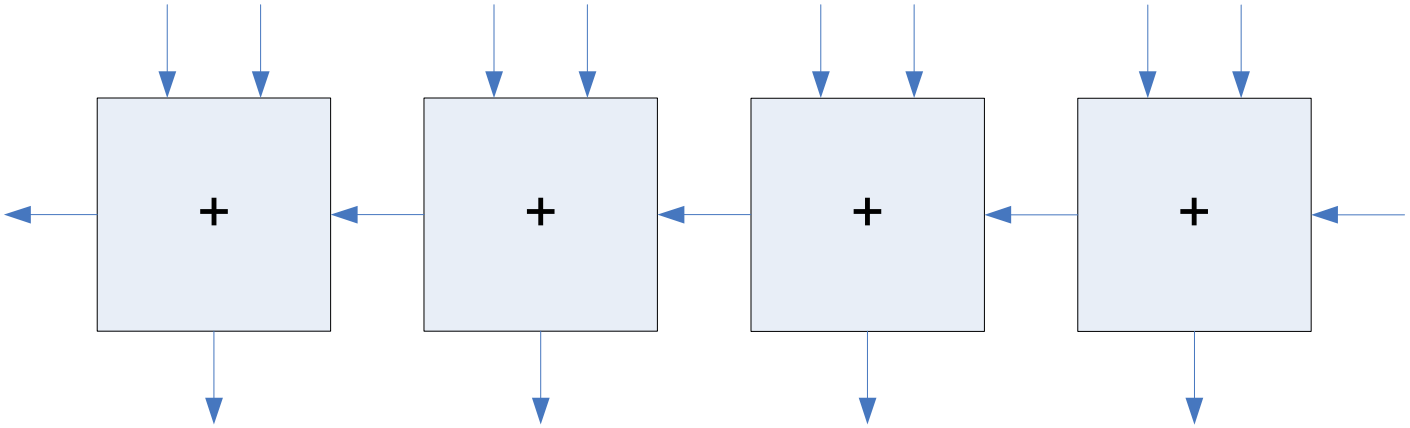
Computer Architecture

Lecture 2

Carry-Lookahead Adder and Memory Elements (Appendix A.6 & A-8)

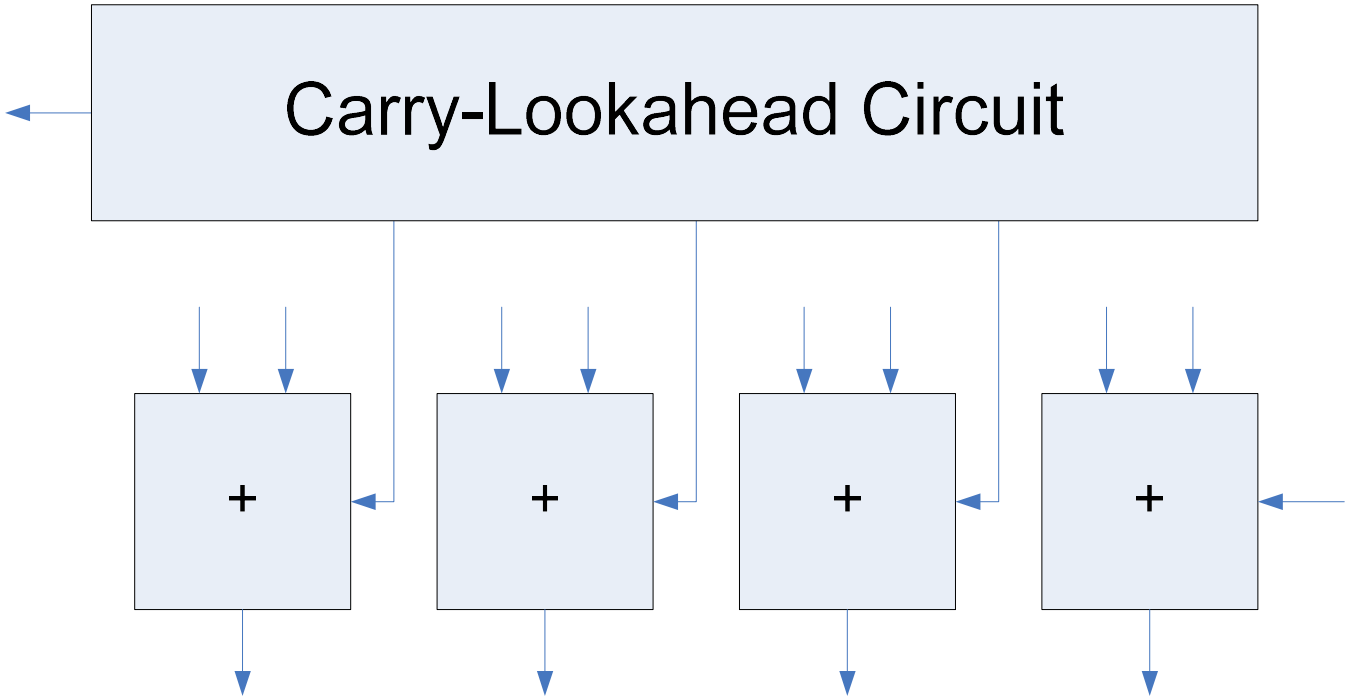
Improving Addition Performance

- The ripple-carry adder is slow



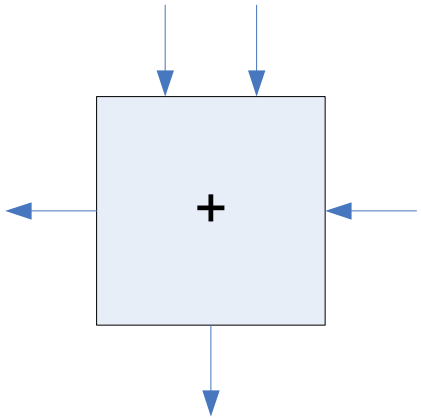
Carry-Lookahead Adder

- Need fast way to find the carry



Carry-Lookahead Adder

- Carry generate and carry propagate



a_i	b_i	g_i	p_i
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

- $g_i = a_i \cdot b_i$
- $p_i = a_i + b_i$

Carry-Lookahead Adder

Carry Equations:

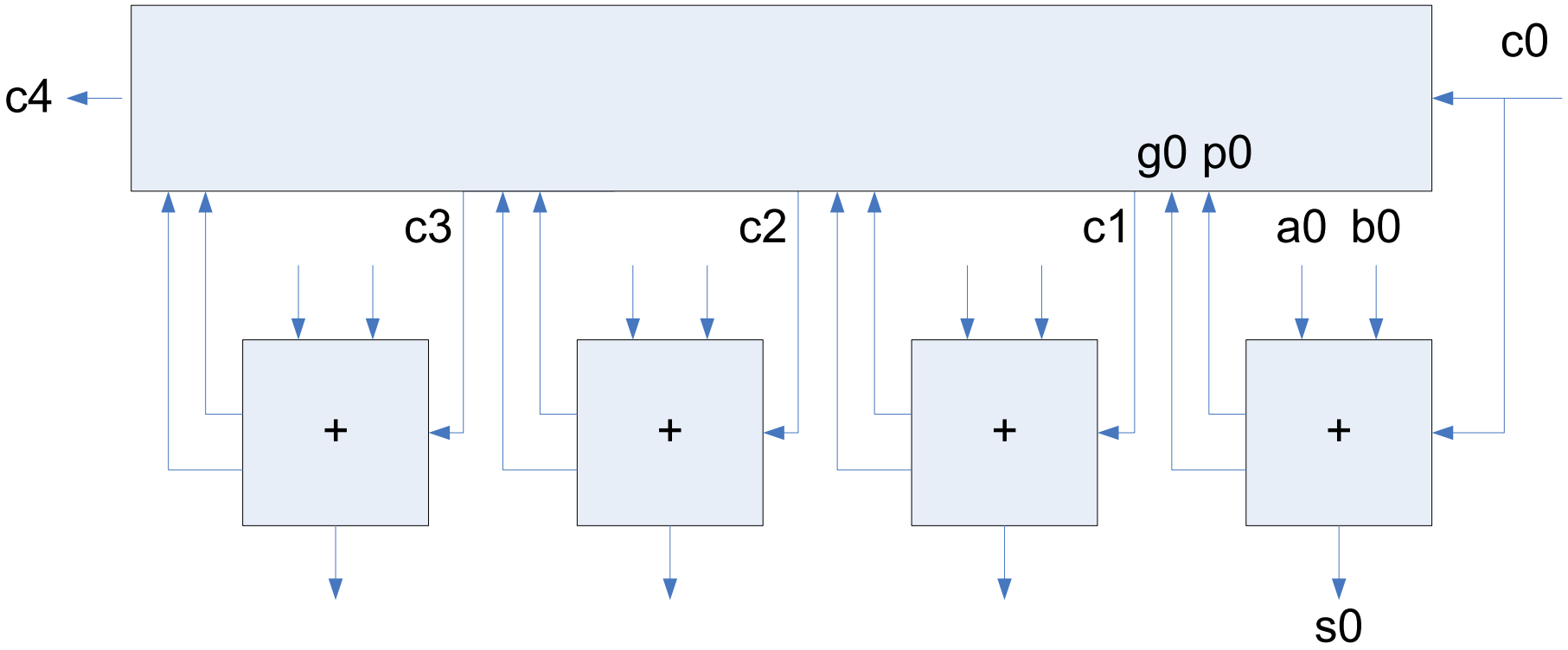
```
c1 = g0 + p0c0


c2 = g1 + p1c1
   = g1 + p1g0 + p1p0c0


c3 = g2 + p2c2
   = g2 + p2g1 + p2p1g0 + p2p1p0c0


c4 = g3 + p3c3
   = g3 + p3g2 + p3p2g1 + p3p2p1g0 + p3p2p1p0c0
```

4-bit Carry-Lookahead Adder

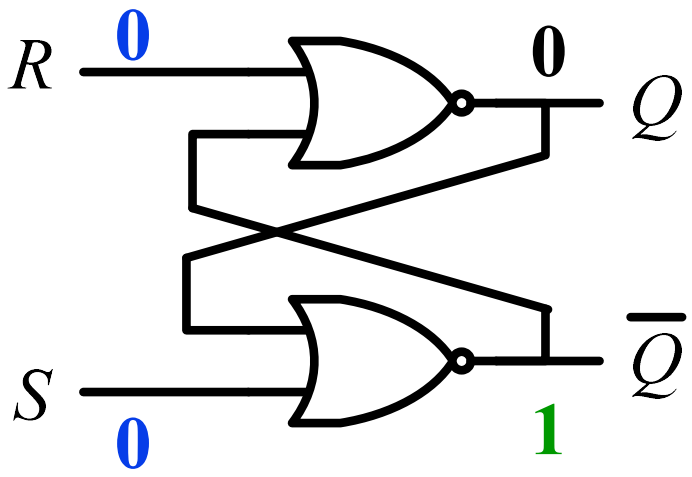


Memory Elements

- Latches
- Flip flops
- Registers

Latches

- *SR* Latch



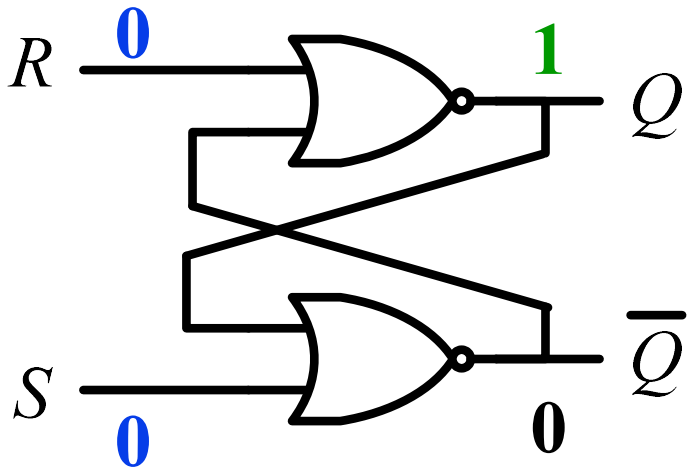
S	R	Q_0	Q	Q'
0	0	0	0	1

$Q = Q_0$

Initial Value

Latches

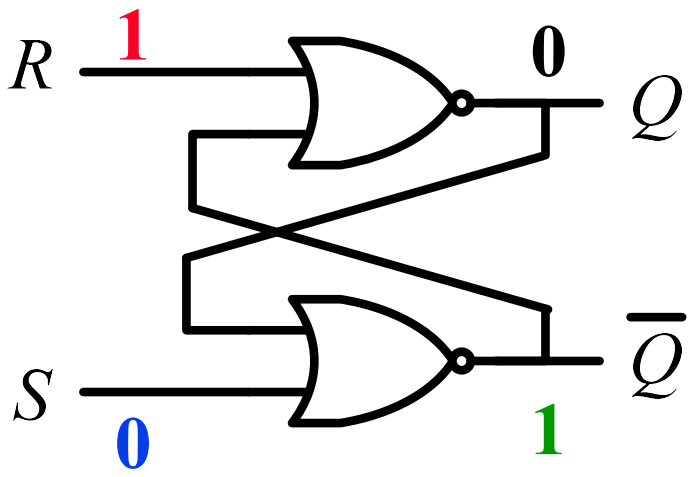
- *SR* Latch



S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	$Q = Q_0$

Latches

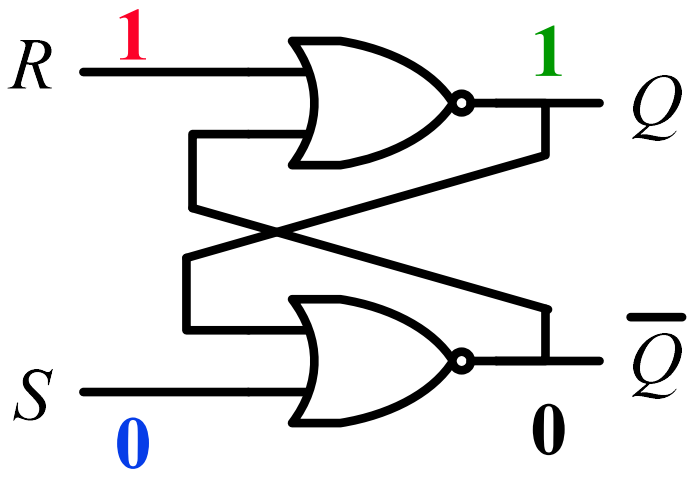
- *SR* Latch



S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$

Latches

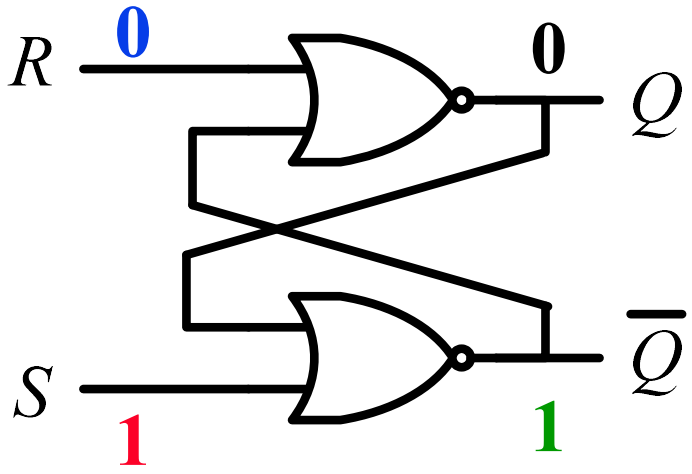
- *SR* Latch



S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	$Q = 0$

Latches

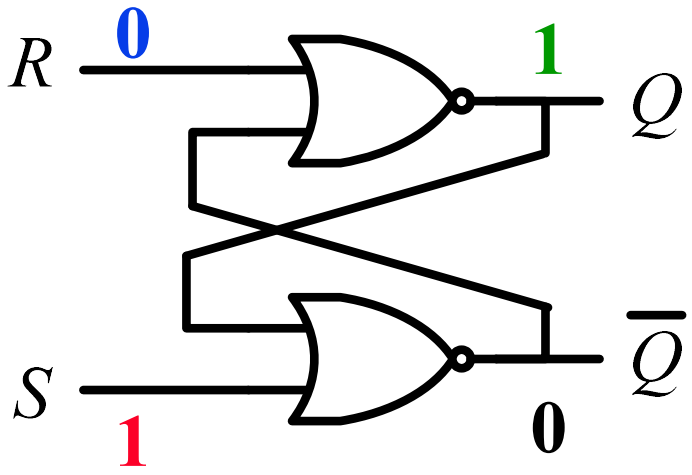
- *SR* Latch



S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$

Latches

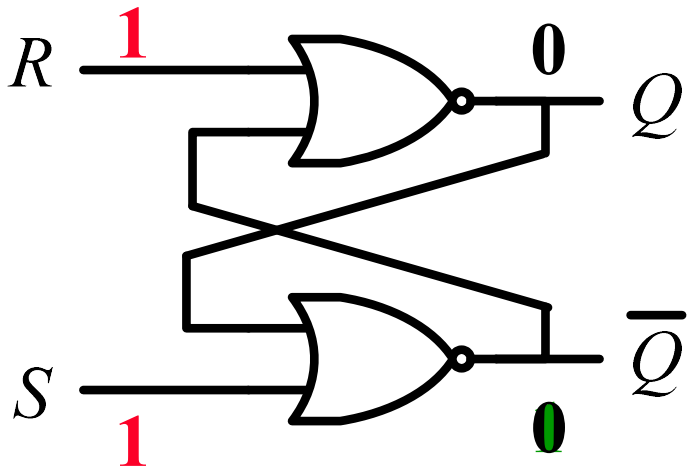
- *SR* Latch



S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$
1	0	1	1	0	$Q = 1$

Latches

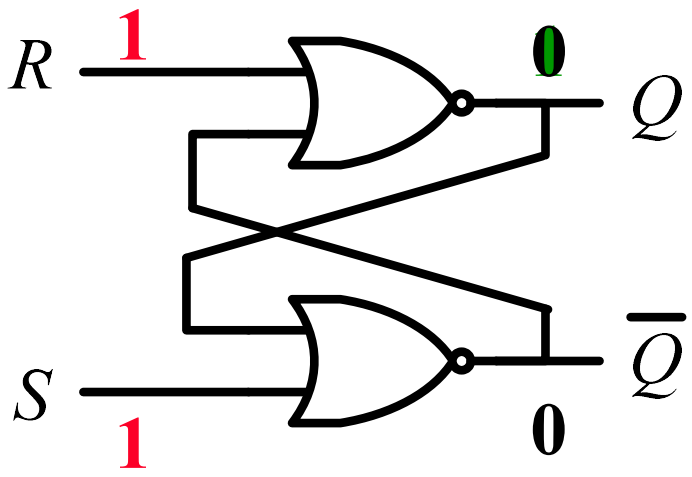
- *SR* Latch



S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$
1	0	1	1	0	
1	1	0	0	0	$Q = Q'$

Latches

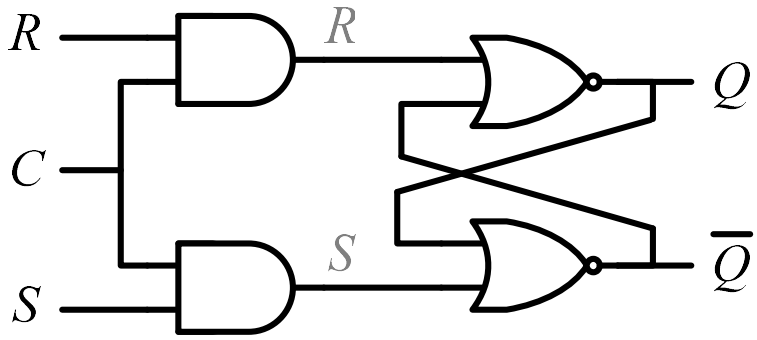
- *SR* Latch



S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$
1	0	1	1	0	
1	1	0	0	0	$Q = Q'$
1	1	1	0	0	$Q = Q'$

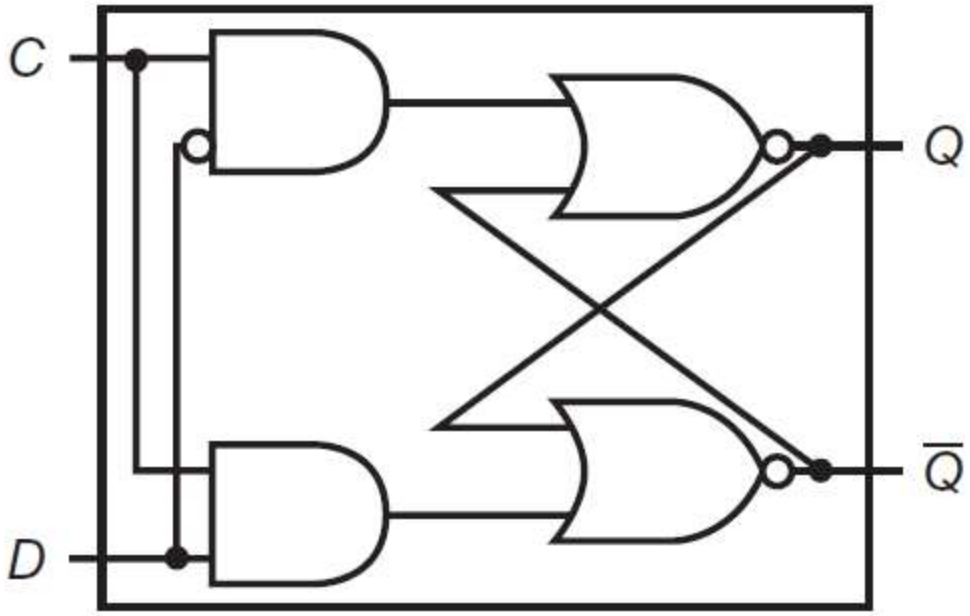
Controlled Latches

- *SR* Latch with Control Input



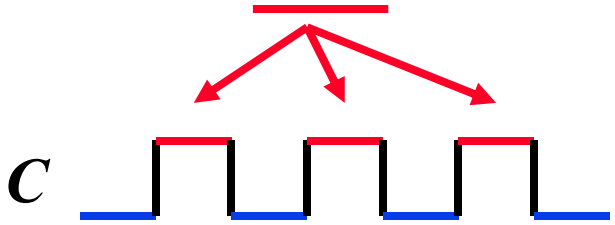
C S R	Q	
0 x x	Q_0	No change
1 0 0	Q_0	No change
1 0 1	0	Reset
1 1 0	1	Set
1 1 1	Q=Q'	Invalid

A D latch implemented with NOR gates

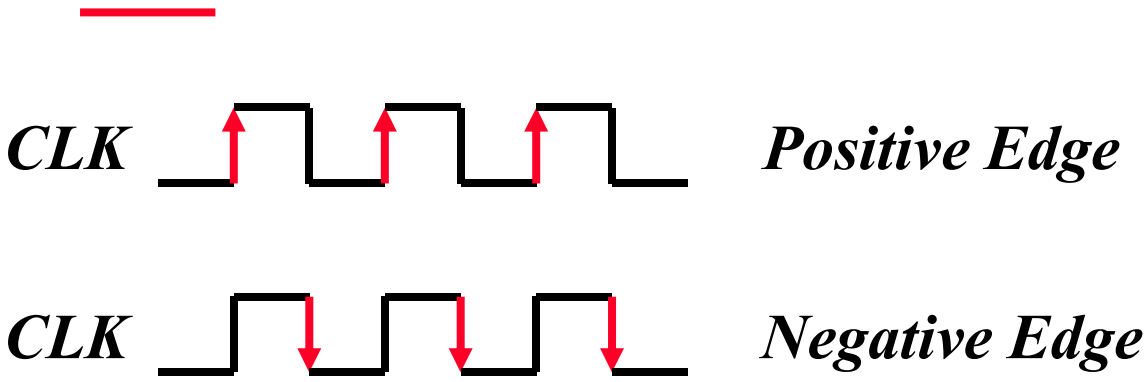


Flip-Flops

- Controlled latches are level-triggered

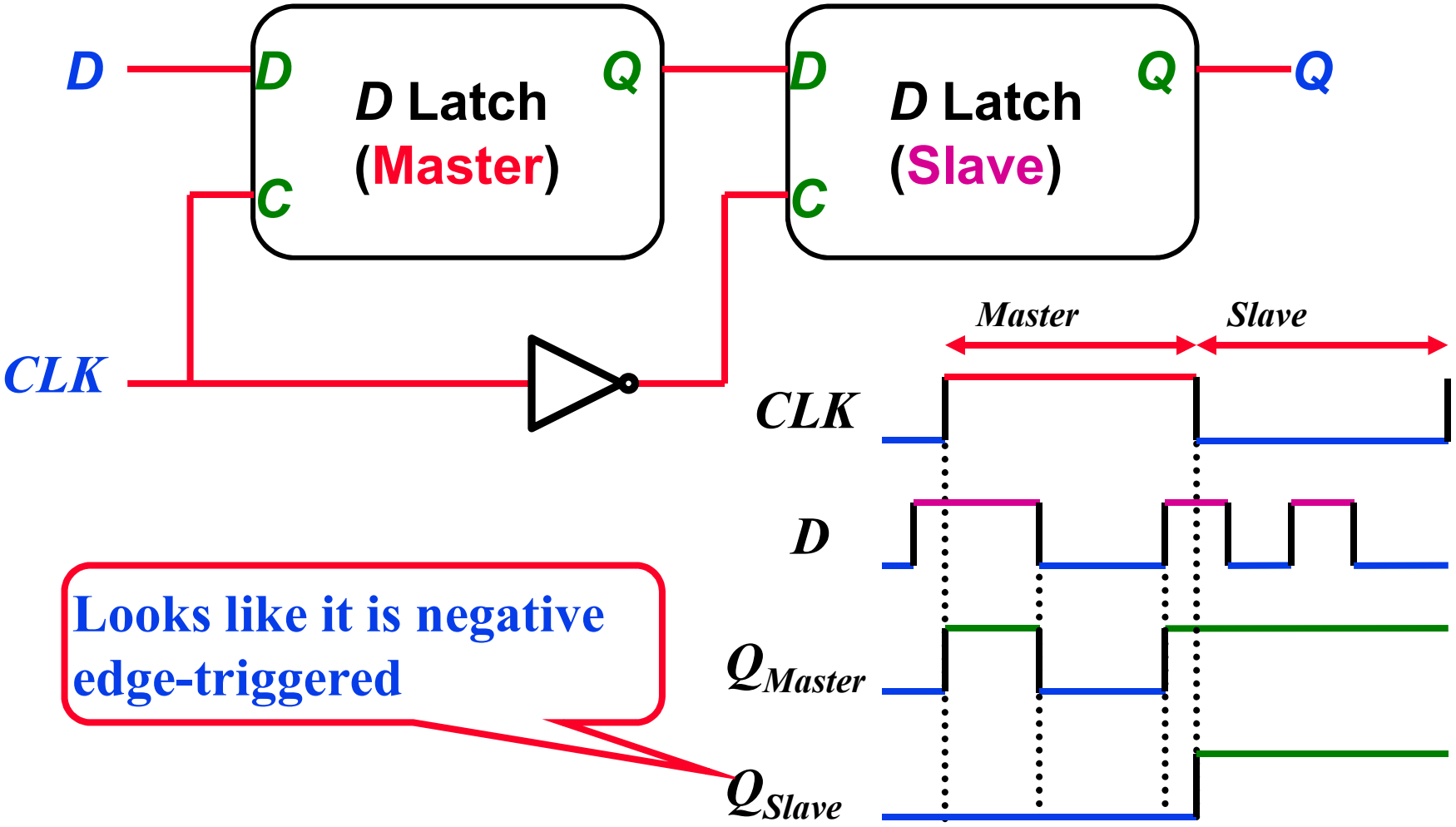


- Flip-Flops are edge-triggered

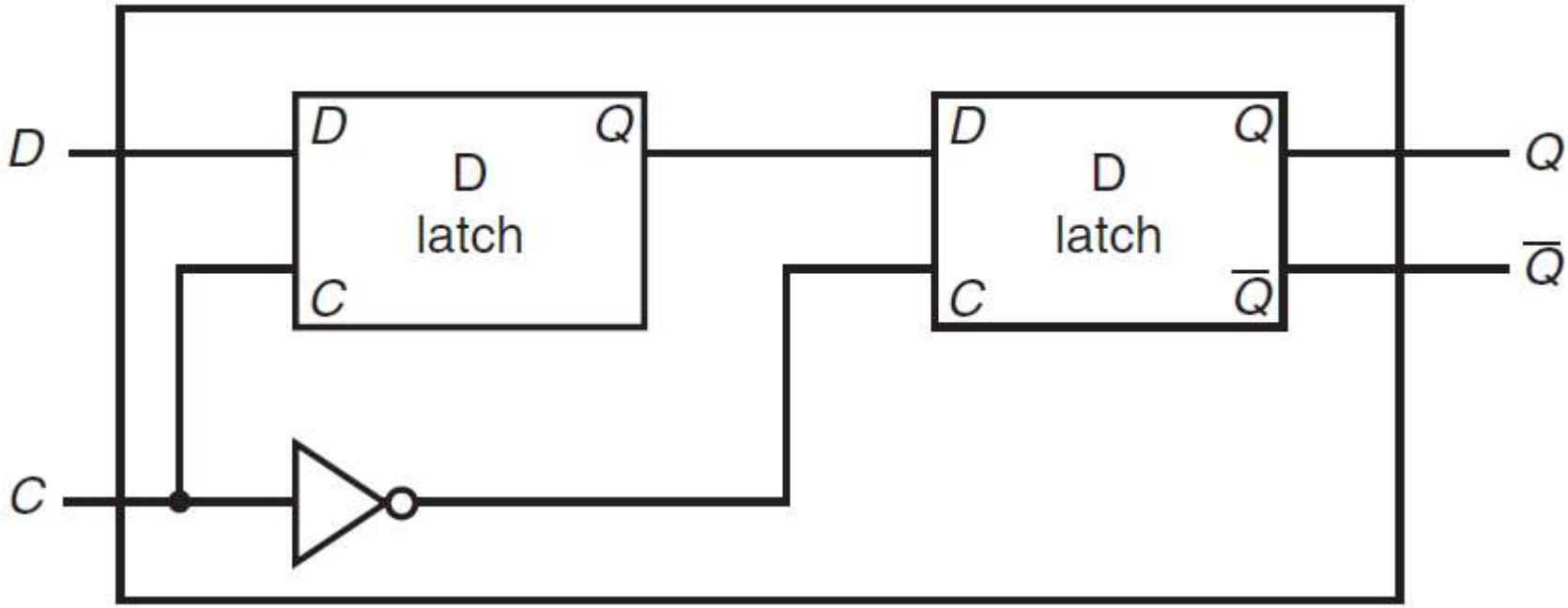


Flip-Flops

- Master-Slave *D* Flip-Flop

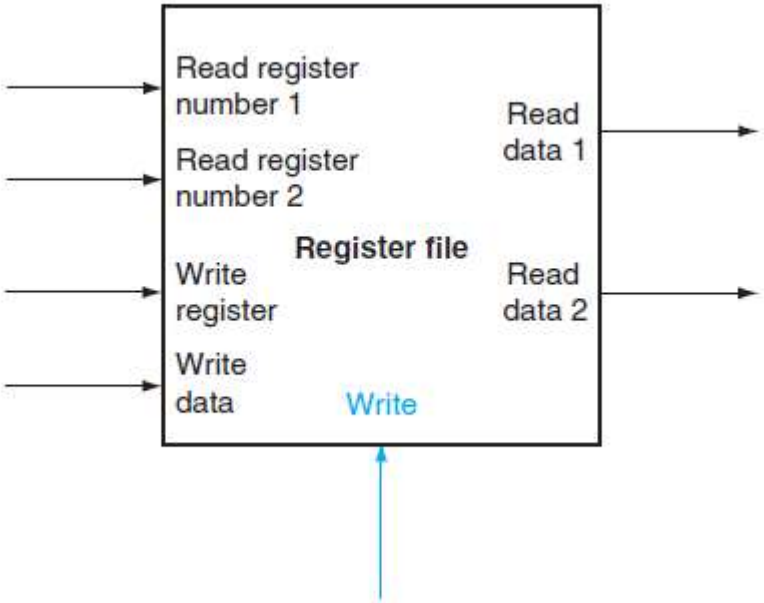


A D flip-flop implemented with a falling-edge trigger



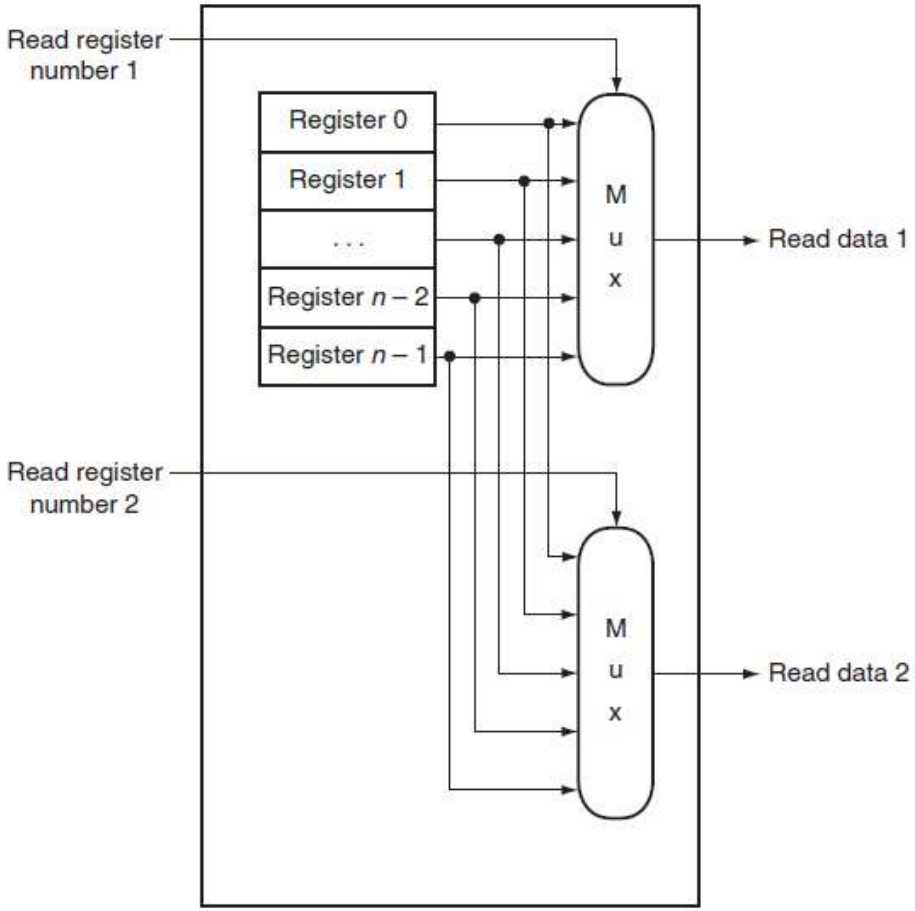
Register files

- A register file with
  - Two read ports
  - One write port
  - Five inputs
  - Two outputs



Read ports of the register file

- An implementation of two register read ports for a 64-bit-wide register file.



Write ports of the register file

- Write port implementation for the register file.