11.1 Introduction

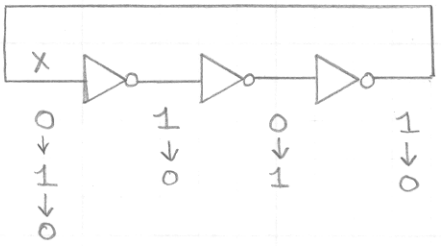
Sequential circuits: output depends on present and past values of inputs (circuit "remembers")

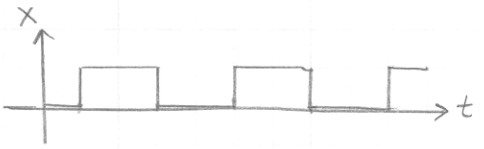
Latches and flip-flops are common memory devices used in sequential circuits

Memory is implemented using feedback with combinational gates:

Odd number of inverters (Lab 2)

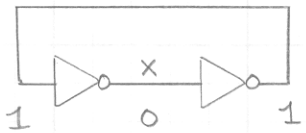
Analyze by assuming value on a node and propagating values to other nodes:



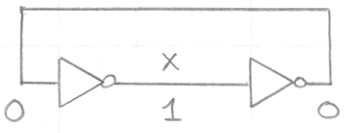


Even number of inverters

Two stable states:



X = 0

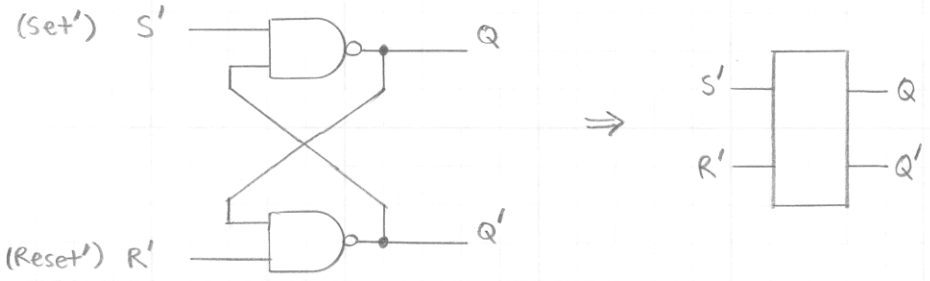


X = 1

Circuit "remembers" previous value (once X=0, X stays 0 until some external action changes it)

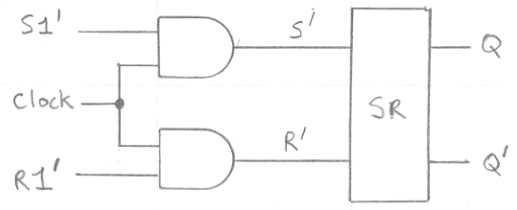
11.2 Set-Reset Latch

SR Latch (NAND Latch)

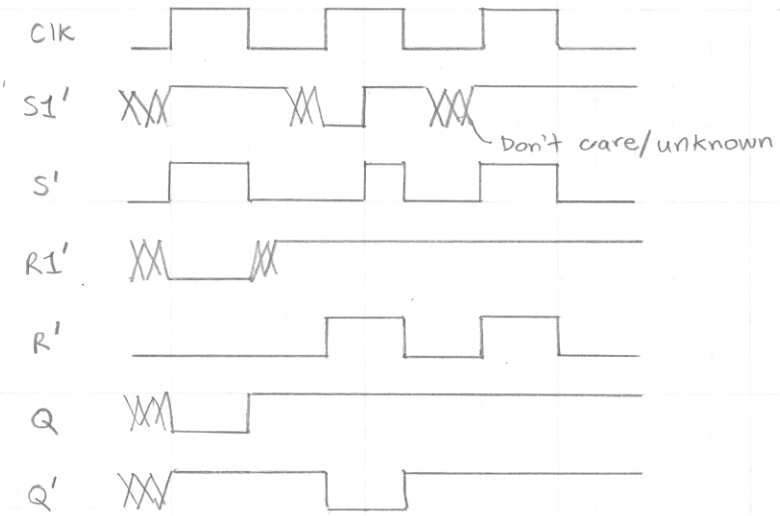


S'	R'	Q	New Q	
0	0	0	1	(Q'=0) } inconsistent / invalid state
0	0	1	1	(Q'=0) }
0	1	0	1	} set
0	1	1	1	}
1	0	0	0	} reset
1	0	1	0	}
1	1	0	0	} no change (remember)
1	1	1	1	}

Clocked SR Latch

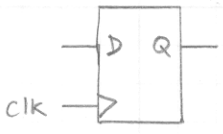


Clock (CIK) ≡ a periodic waveform used to control sequential circuit timing

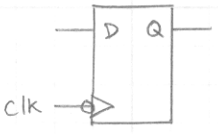


11.4 Edge Triggered D Flip-Flop

D Flip-Flop: has two inputs, D (data) and CIK (clock). Output only changes in response to transitions on CIK



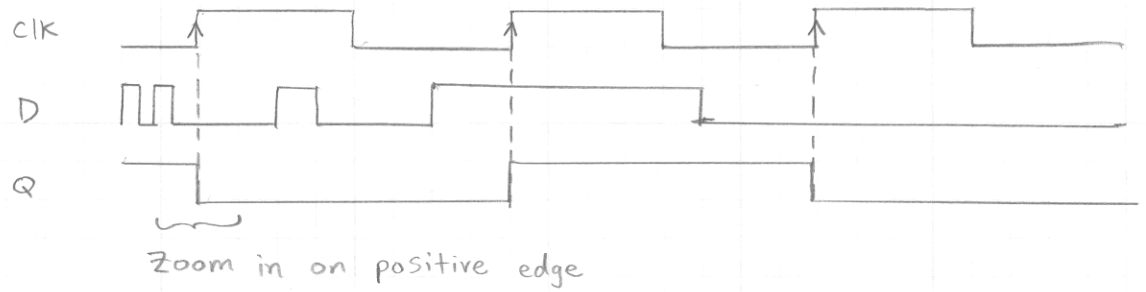
Rising Edge Triggered



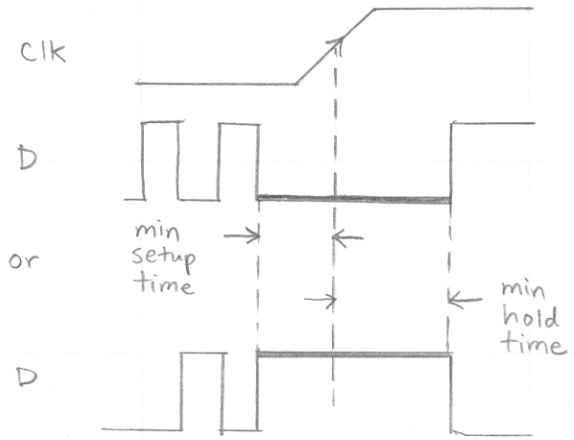
Falling Edge Triggered

D	Q	New Q
0	0	0
0	1	0
1	0	1
1	1	1

Truth Table



Zoom in on positive edge



Setup Time: input must not change before clock edge

Hold Time: input must not change after clock edge

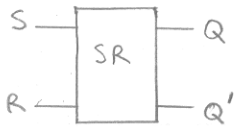
Setup and hold time constraints must be met for proper circuit operation.

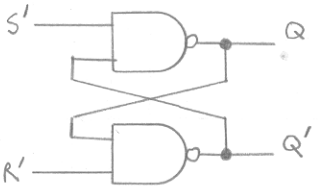
Latches and Flip-Flops Review

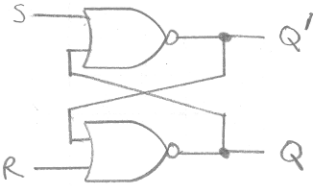
Flip-flop: Memory element whose output changes on the active edge of a special input (clk)

Latch: Memory element whose output changes on the active level of a special input (e.g., gated latch) or at any time (e.g., SR latch)

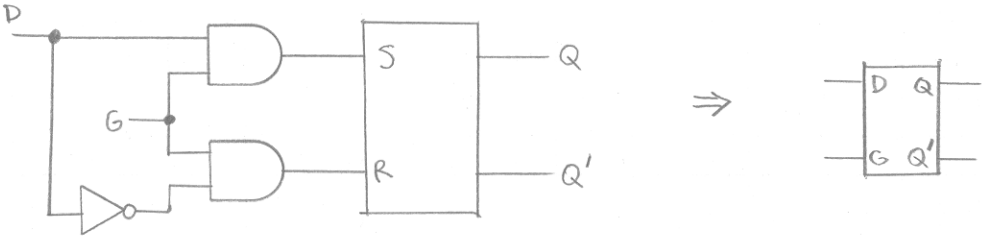
SR Latches



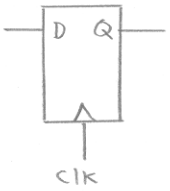




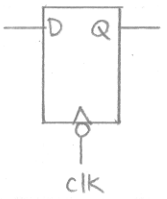
Gated Latches (Transparent Latches)



D Flip-Flop



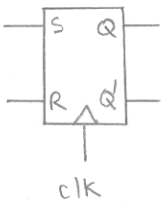
a.) positive edge-triggered



b.) negative edge-triggered

Must meet setup and hold time requirements for correct operation

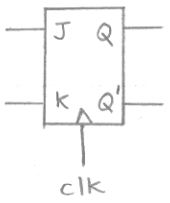
SR Flip-Flop



Q^+ ≡ next Q state

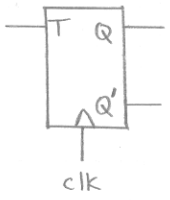
S	R	Q^+	
0	0	memory	$(Q+ = Q)$
0	1	reset	$(Q+ = 0)$
1	0	set	$(Q+ = 1)$
1	1	—	invalid

JK Flip-Flop



J	K	Q+	
0	0	memory	(Q+ = Q)
0	1	reset	(Q+ = 0)
1	0	set	(Q+ = 1)
1	1	toggle Q	(Q+ = Q', i.e. 0→1 or 1→0)

T Flip-Flop



T	Q+	
0	memory	(Q+ = Q)
1	toggle Q	(Q+ = Q')

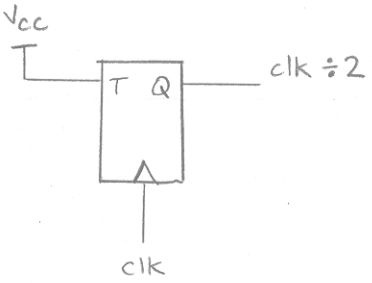
Other Inputs

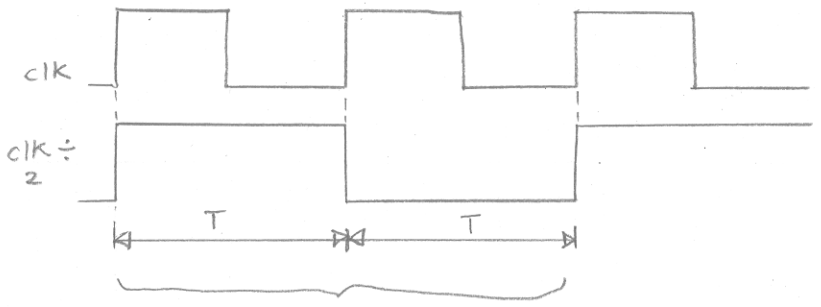
Asynchronous: without regard to the clock signal
Synchronous: effective only on an active clock edge

Ex: Four combinations to initialize value stored in a flip-flop:

	Clear (reset)	Set
Asynchronous	X	X
Synchronous	X	X

Ex: Flip-flops can be used to <u>divide</u> a clock (generate a clock at a lower frequency):





Old period = T, New period = 2T

We can cascade multiple T flip-flops to produce clk ÷ 4, clk ÷ 8, etc.
Lab 6 uses clock division on the Altera board.