2.2. Explain what are the differences between latches and flip flops with your own words.
The main difference between a latch and a flip flop is that the first is asynchronous, and the second is synchronized with a clock. When, for instance, a set signal is given to latch, the time required to switch states will only depend on the internal delays of gate. Flip flops, however, are triggered either by positive or negative edge of clock signal, according to their type. Implementation of synchronization requires more logic gates.

2.4. Construct the truth table of an SR latch which does not have an Enable input.

S	R	$Q(t+1)$	$\overline{Q}(t+1)$	
0	0	$Q(t)$	$\overline{Q}(t)$	(preserve)
1	0	1	0	(set)
0	1	0	1	(reset)
1	1	x	x	(forbidden)

where $Q(t)$ and $Q(t+1)$, $\overline{Q}(t)$ and $\overline{Q}(t+1)$ represent previous and next states, and x is unknown.

2.6. Construct the truth table of a D flip flop.

D	CLK	$Q(t+1)$	$\overline{Q}(t+1)$	
ϕ	ϕ	$Q(t)$	$\overline{Q}(t)$	(preserve)
1	$\_\!\!\sqcap$	1	0	(set)
0	$\_\!\!\sqcap$	0	1	(reset)

where $Q(t)$ and $Q(t+1)$, $\overline{Q}(t)$ and $\overline{Q}(t+1)$ represent previous and next states, ϕ is any state and $\_\!\!\sqcap$ symbol means rising edge, i.e. transition from low to high.

Part 1:

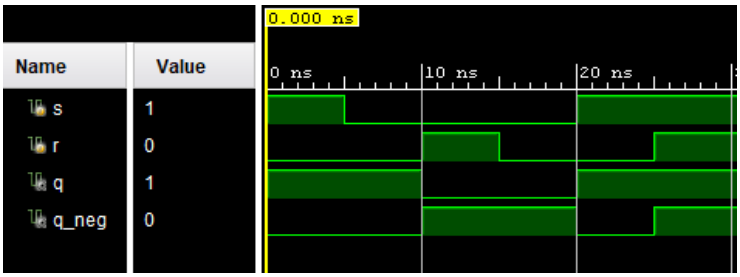

Figure 1: SR Latch simulation

The behaviour of SR Latch can be expressed with an equation of form $Q(t+1) = f(S; R; Q(t))$. From the truth table, all combinations of S and R are selected except the forbidden state, and $Q(t+1)$ is expressed and simplified using rules of Boolean algebra as follows:

$$
\begin{aligned}
Q(t+1) &= \bar{S}\bar{R}Q(t) + \bar{S}R \cdot 0 + S\bar{R} \cdot 1 && \text{(Truth table)} \\
&= \bar{S}\bar{R}Q(t) + 0 + S\bar{R} \cdot 1 && \text{(Dominance)} \\
&= \bar{S}\bar{R}Q(t) + S\bar{R} && \text{(Identity)} \\
&= \bar{S}\bar{R}Q(t) + S\bar{R} + \bar{R}\bar{R}Q(t) && \text{(Consensus theorem)} \\
&= \bar{S}\bar{R}Q(t) + S\bar{R} + \bar{R}Q(t) && \text{(Idempotency)} \\
&= Q(t)(\bar{S}\bar{R} + \bar{R}) + S\bar{R} && \text{(Distributivity)} \\
&= Q(t)\bar{R} + S\bar{R} && \text{(Absorption)} \\
&= Q(t)\bar{R} + S
\end{aligned}
$$

In the final step, $\bar{R}$ is omitted because state $S = 1, R = 1$ is forbidden and should never happen. Therefore omitting it does not change the truth table. To summarize, SR Latch's output is 1 whenever S is on and R is off, is 0 whenever R is on and S is off, and it maintains the previous state if both S and R go low. It is therefore called a memory circuit.

Part 3:

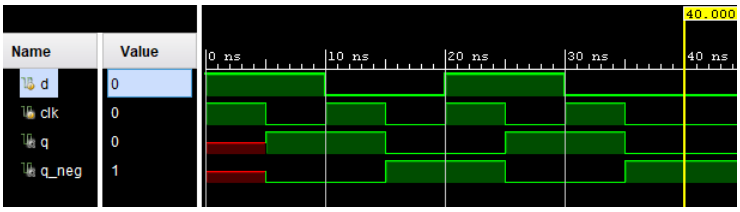

Figure 2: Negative edge triggered D flip-flop simulation

As it is seen from the wave form, D flip-flop can change state only on the falling clock edge.

Part 6:

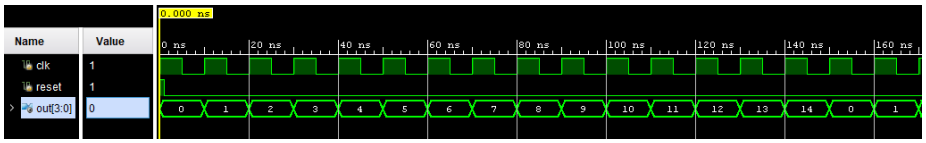

Figure 3: Synchronous up counter simulation

The truth table is as follows:

CLK	Reset	$\{A_3, A_2, A_1, A_0\}$	$\{A_3, A_2, A_1, A_0\}^+$
ϕ	⎍	$\{\phi, \phi, \phi, \phi\}$	$\{0, 0, 0, 0\}$
⎍	0	$\{0, 0, 0, 0\}$	$\{0, 0, 0, 1\}$
⎍	0	$\{0, 0, 0, 1\}$	$\{0, 0, 1, 0\}$
⎍	0	$\{0, 0, 1, 0\}$	$\{0, 0, 1, 1\}$
⎍	0	$\{0, 0, 1, 1\}$	$\{0, 1, 0, 0\}$
⎍	0	$\{0, 1, 0, 0\}$	$\{0, 1, 0, 1\}$
⎍	0	$\{0, 1, 0, 1\}$	$\{0, 1, 1, 0\}$
⎍	0	$\{0, 1, 1, 0\}$	$\{0, 1, 1, 1\}$
⎍	0	$\{0, 1, 1, 1\}$	$\{1, 0, 0, 0\}$
⎍	0	$\{1, 0, 0, 0\}$	$\{1, 0, 0, 1\}$
⎍	0	$\{1, 0, 0, 1\}$	$\{1, 0, 1, 0\}$
⎍	0	$\{1, 0, 1, 0\}$	$\{1, 0, 1, 1\}$
⎍	0	$\{1, 0, 1, 1\}$	$\{1, 1, 0, 0\}$
⎍	0	$\{1, 1, 0, 0\}$	$\{1, 1, 0, 1\}$
⎍	0	$\{1, 1, 0, 1\}$	$\{1, 1, 1, 0\}$
⎍	0	$\{1, 1, 1, 0\}$	$\{0, 0, 0, 0\}$

Part 7: To implement pulse width modulation with a circular shift register, there must be a multiplexer which will choose between the input wave pattern and output of previous flip flop, and output the signal as input to the next flip flop. When load flag is low, flip flops behave like a circular shift register. When load flag becomes high, circular transmission is stopped and all flip flops are written to. This is the design of 2:1 multiplexer we used:

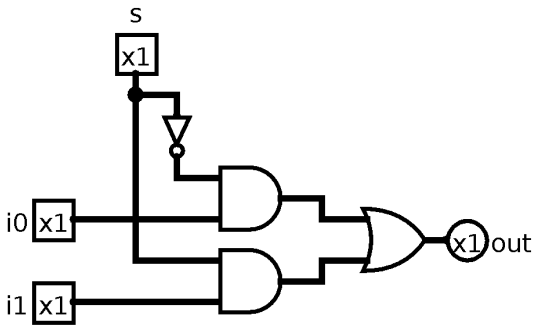

Figure 4: 2:1 multiplexer design in Logisim

It was also described and simulated with Verilog:

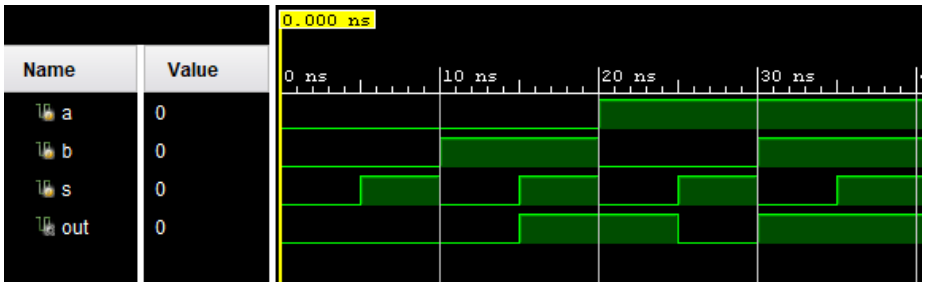

Figure 5: 2:1 multiplexer simulation in Verilog

The pulse generator itself is too long to fit on the page, and with that reason the following diagram is it's smaller version with same behaviour. Because D flip flops are negative edge triggered, the clock is inverted in order to make it a positive edge triggered device:

In the Verilog description we prepared, pulse generator has a 16-bit input as required. The result of simulation confirms it can deliver various pulses, such as half, quarter, one eigth of clock frequency and signals with 1/7, 3/13, 11/5 pulse-gap duration rate:

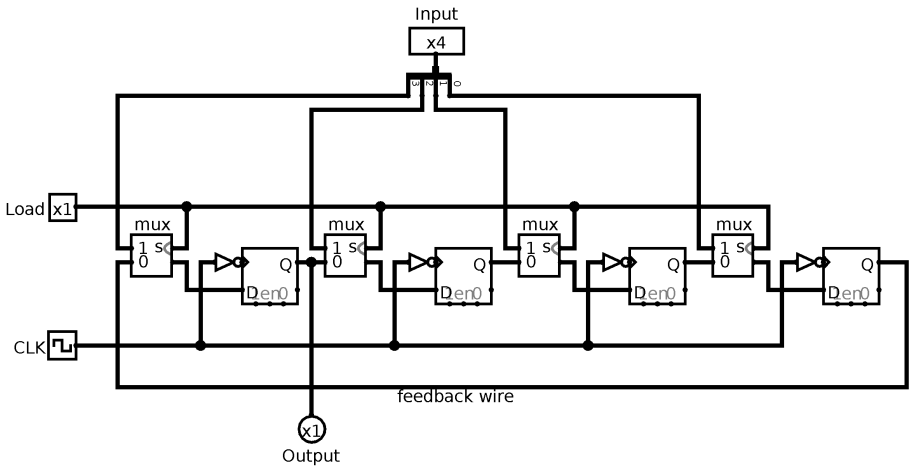

Figure 6: Pulse generator design in Logisim

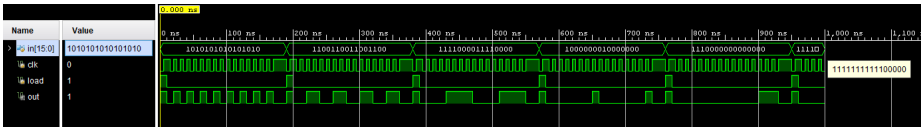

Figure 7: Pulse generator simulation in Verilog