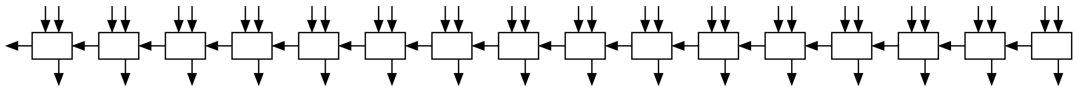
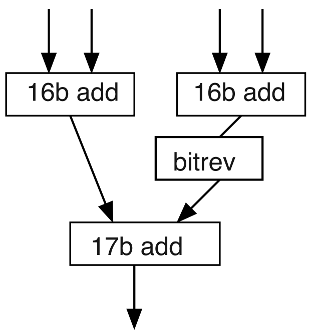
Consider a 16b (and 17b) ripple-carry adder built out of 1-bit adder slices. The delay from the three inputs of the 1-bit adder to the two outputs is T_{bit}.



1. What is the worst-case delay through the 16b adder?
2. After a change in an input, what is the **shortest** amount of time before an output bit changes?
3. Consider the somewhat unusual 4-input adder tree shown. Assume negligible ($<< T_{bit}$) delay in bitrev. (bitrev: Out[i]=In[17-i])



What happens if we provide new set of inputs to this circuit every $20T_{bit}$?

4. How might we implement a latch that behaves as follows:

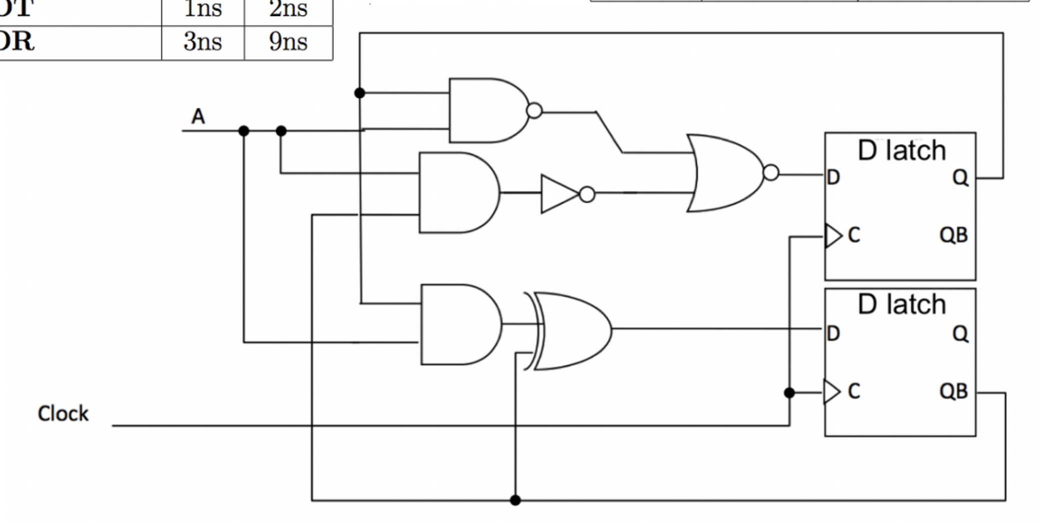
```
if (φ==1)
    Out=/In
else
    Out=Out
```

(a) Using combinational logic (at gate level. Hint: Think about a mux...)?

(b) Using Pass transistors?

	Min	Max
OR/AND	3ns	5ns
NOR/NAND	2ns	4ns
NOT	1ns	2ns
XOR	3ns	9ns

Latch:		Min	Max
	Clock to Q	2ns	3ns
	Setup time	7ns	
	Hold time	6ns	



5. Timing Constraints:

$$T \geq t_{c-q} + t_{plogic} + t_{su} \quad (1)$$
$$t_{cdregister} + t_{cdlogic} \geq t_{hold} \quad (2)$$

(a) What is the minimum clock period, T, that ensures correct operation?

(b) Add inverter pairs to the design above such that there are no hold time variations.