

CPS 104
Computer Organization and Programming
Lecture 11: Gates, Buses, Latches.

Robert Wagner

Overview of Today's Lecture:

- **The MIPS ALU**
- **Shifter**
- **The Tristate driver**
- **Bus Interconnections**
- **Register Cell**
- **The Register File**

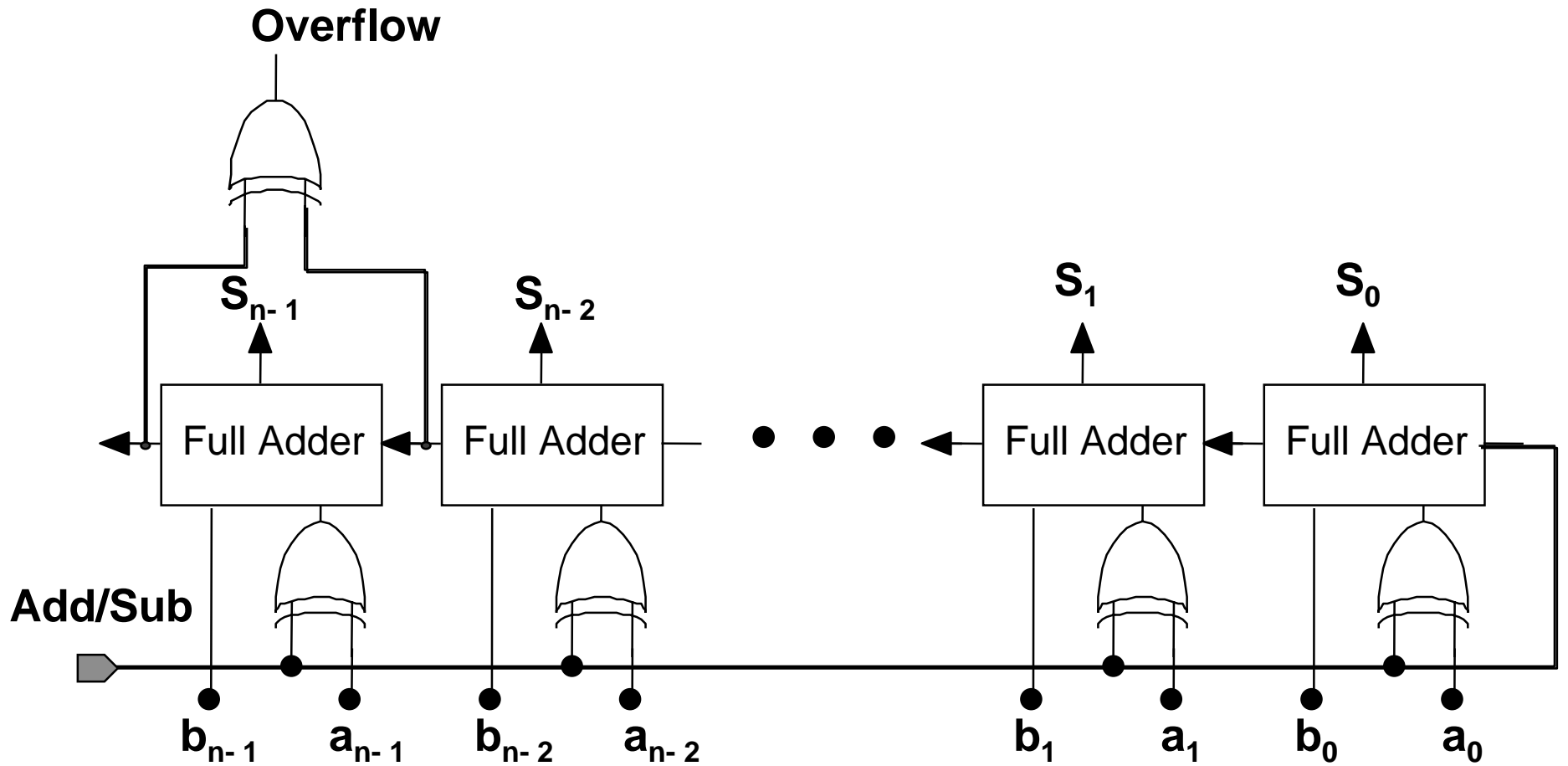
Read Appendix B

Overflow Detection and Carries

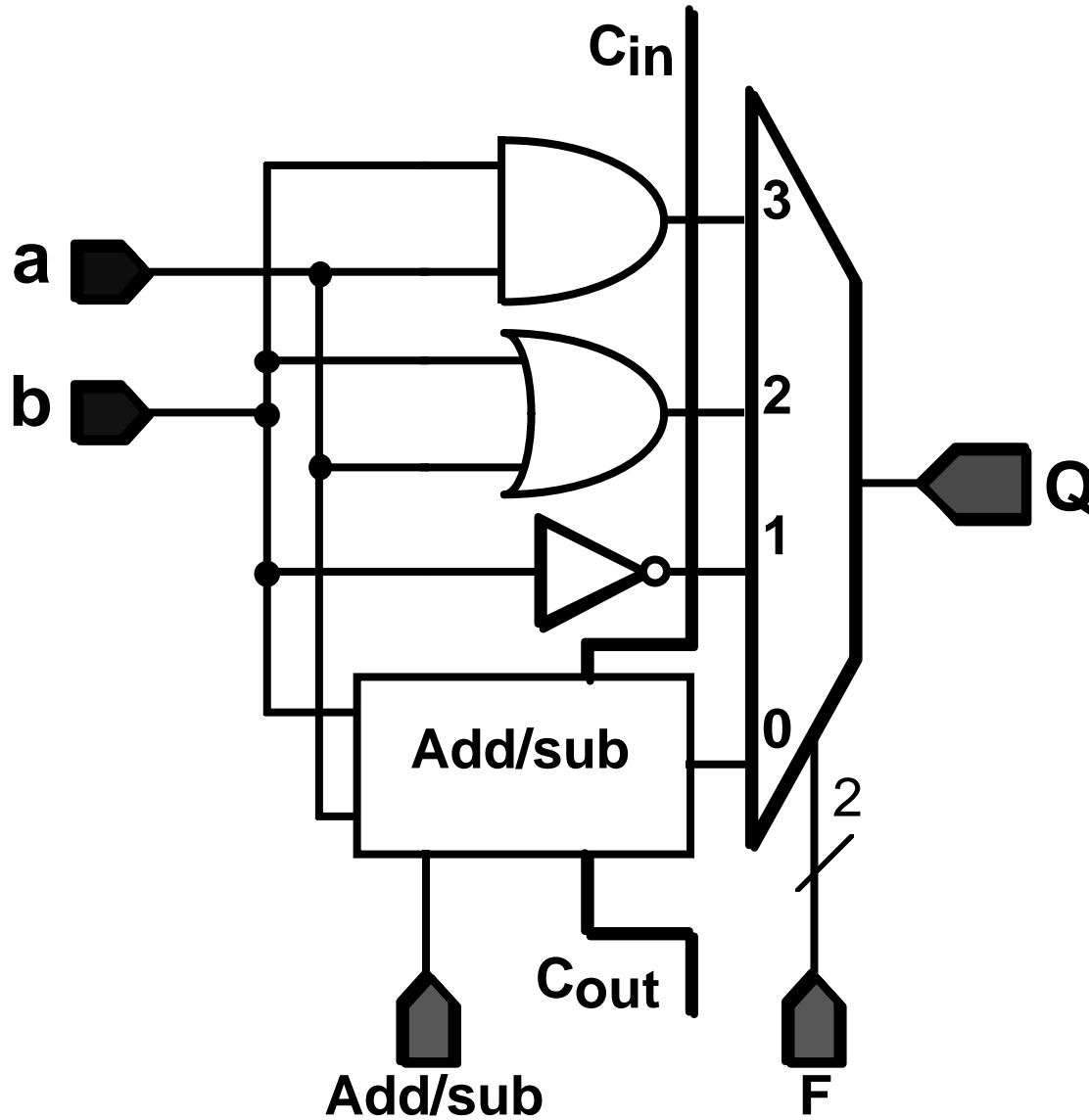
- In Addition and Subtraction operations, Overflow occurs when two number of same sign are added, and result has different sign.
- Must be a function of what happens in high-order full-adder

A_{31}	B_{31}	Cin_{31}	S_{31}	$Cout_{31}$	OVF	Cin= Cout
0	0	0	0	0	F	Y
0	0	1	1	0	T	N
0	1	0	1	0	F	Y
0	1	1	0	1	F	Y
1	0	0	1	0	F	Y
1	0	1	0	1	F	Y
1	1	0	0	1	T	N
1	1	1	1	1	F	Y

Add/Subtract With Overflow detection



MIPS ALU Slice

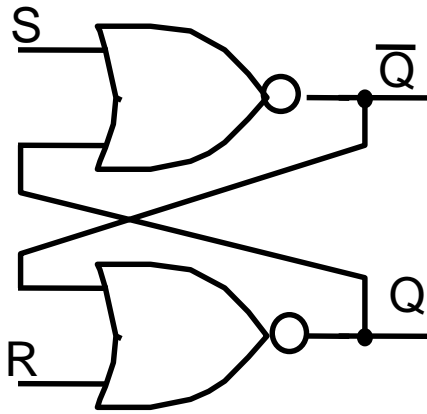


A	F	Q
0	0	a + b
1	0	a - b
-	1	NOT b
-	2	a OR b
-	3	a AND b

Memory Elements

- **All the circuit we looked at so far are combinational circuits: the output is a Boolean function of the inputs.**
- **We need circuits that can remember values. (registers)**
- **The output of the circuit is a function of the input AND a function of a stored values (state) .**
- **Circuits with memory are called sequential circuits.**

Reset-Set Latch

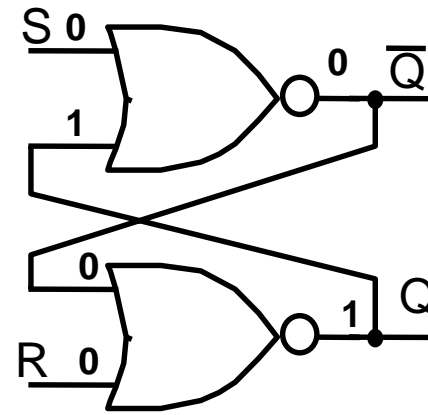
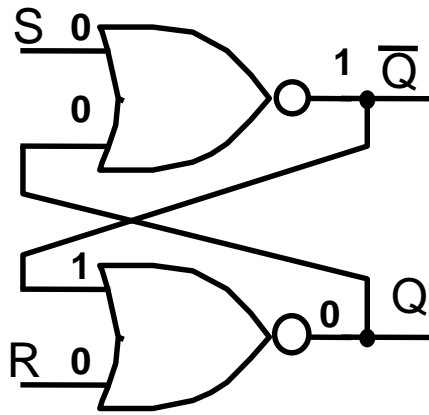


R	S	A(Q)	E(Q)
F	F	F	FS
F	F	T	TS
F	T	F	TU
F	T	T	TS
T	F	F	FS
T	F	T	FU
T	T	F	F?
T	T	T	FU

R	S	Q
F	F	Q
F	T	T
T	F	F
T	T	-

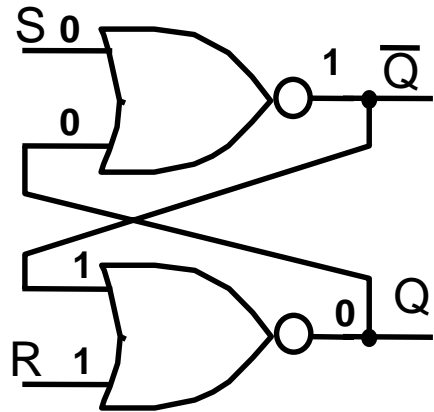
- Pick a wire in each cycle, call it a “variable”. (Here, Q).
- A(Q) = “Activation of Q” = current value of Q. Treat as input.
- E(Q) = “Excitation of Q” = next future value of Q. Treat as output.
- When A(Q)=E(Q), circuit is “stable” = doesn’t change state
- When A(Q)≠E(Q), circuit state goes from A(Q) to E(Q)

Reset-Set Latch (cont.)

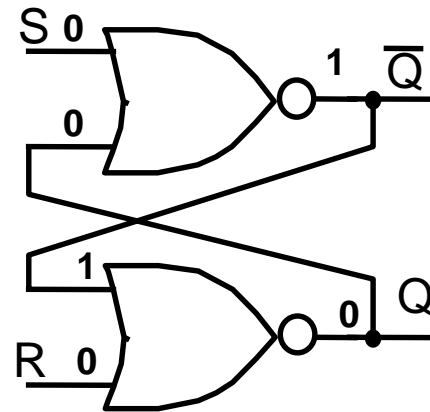


With $S=0$ and $R=0$, BOTH states of wire Q are stable.

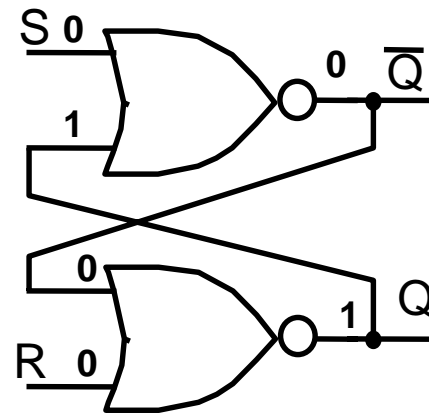
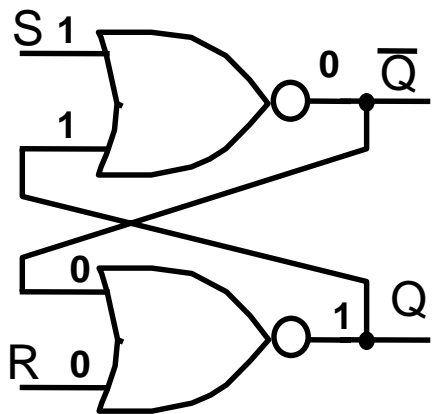
Reset-Set Latch (cont.)



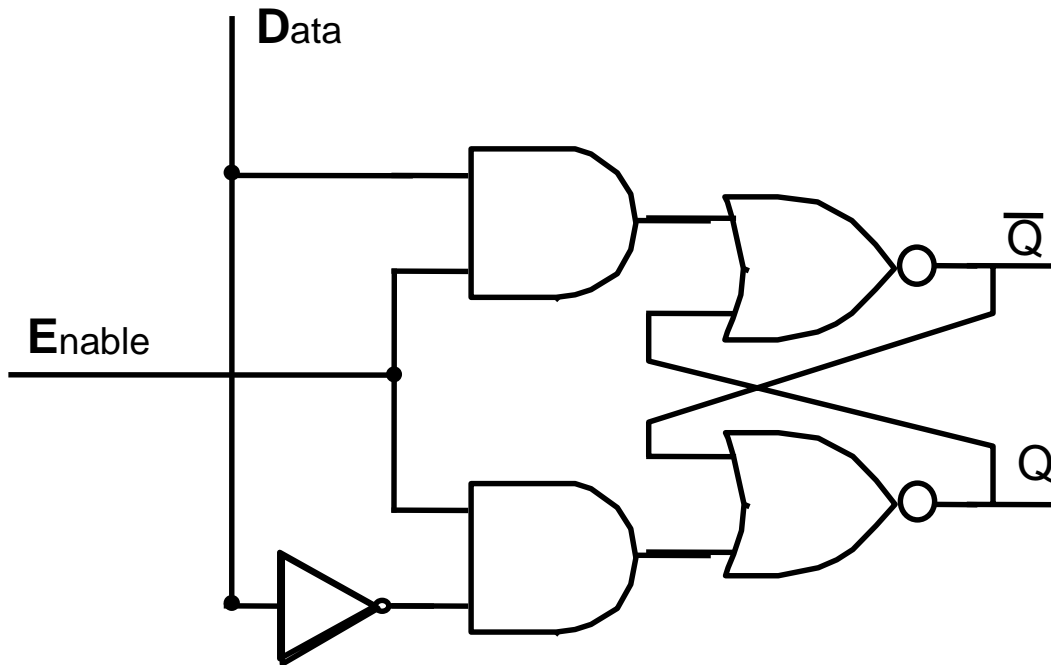
R
e
s
e
t



S
e
t



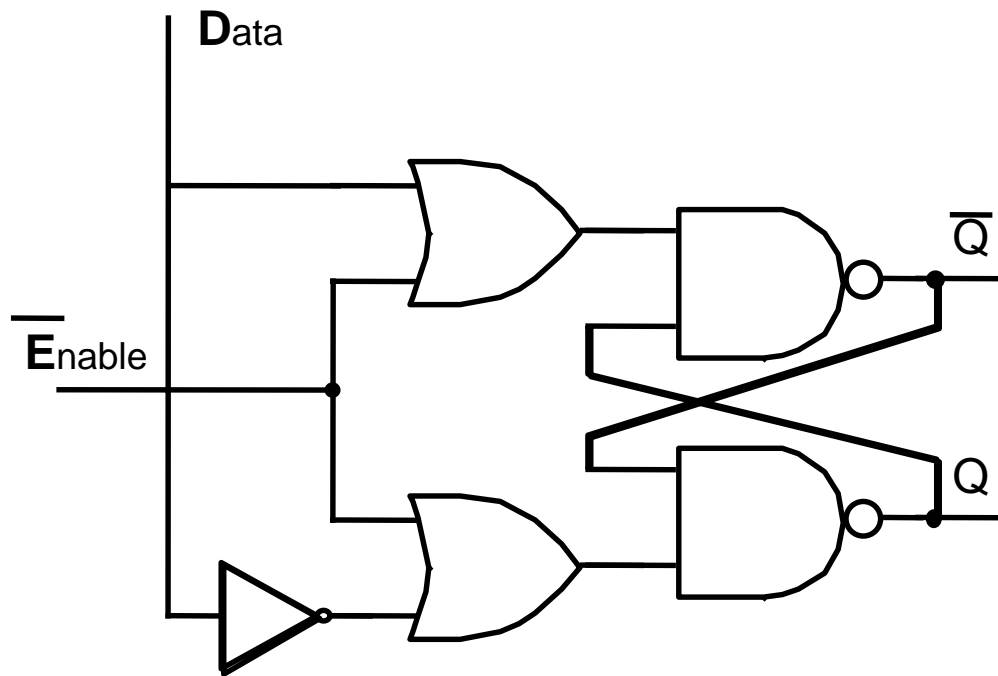
Data-Latch



D	E	Q
0	1	0
1	1	1
-	0	Q

Negative Edge D-Latch

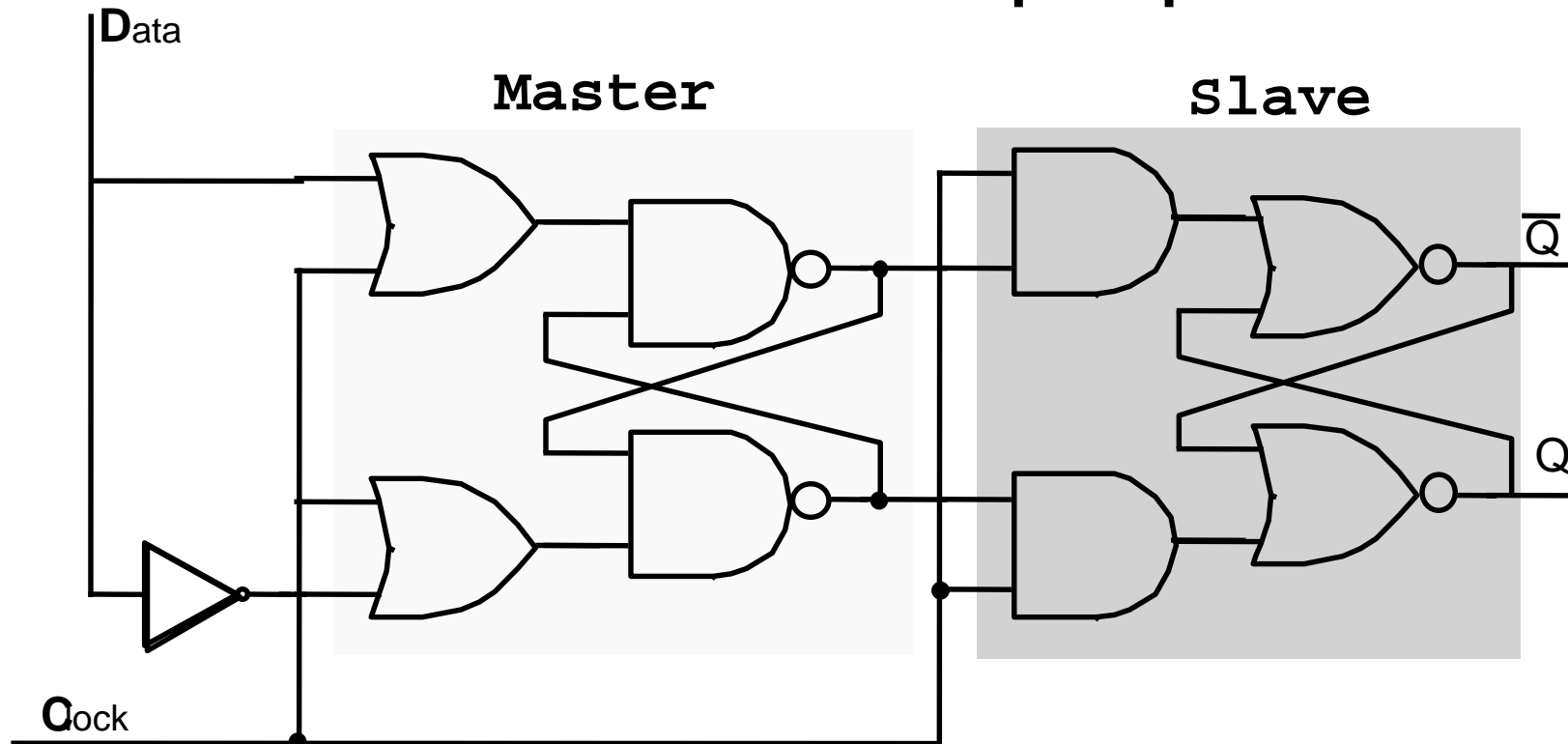
Data-Latch (cont.)



D	\overline{E}	Q
0	1	0
1	1	1
-	0	Q

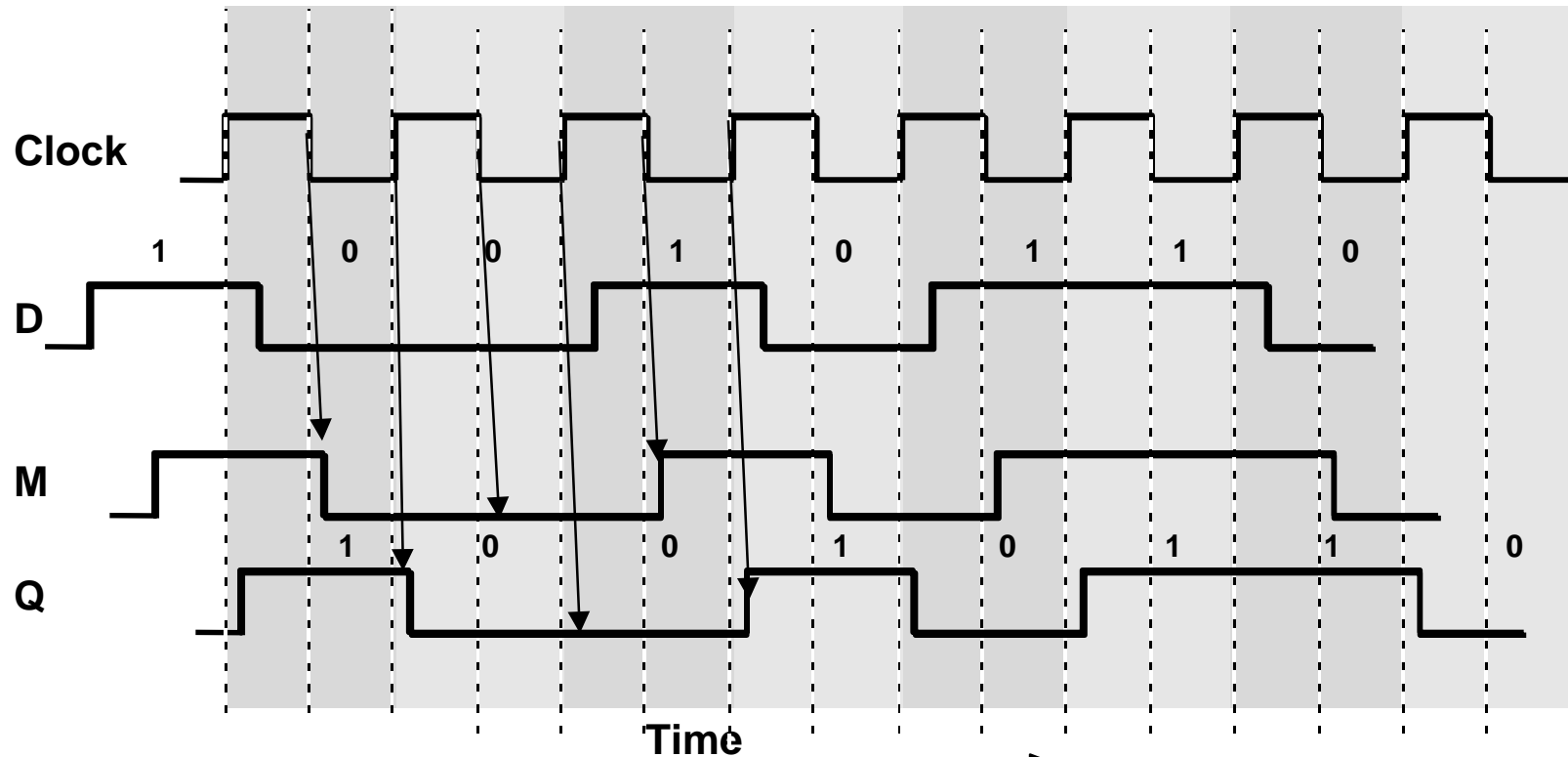
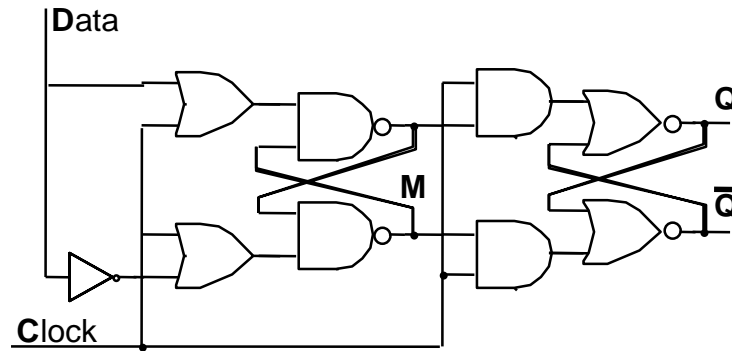
Positive Edge D-Latch

Master-Slave Data-Flip-Flop



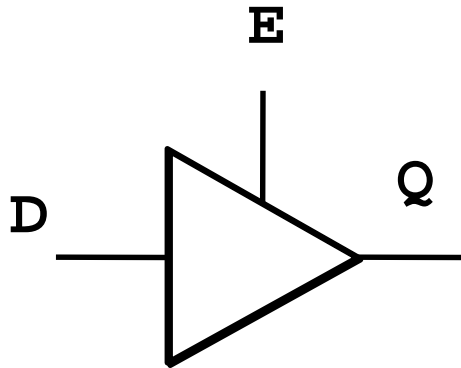
- While Clock false (low) D is copied to the master stage and the slave is stable.
- On Clock \uparrow the Master stage is transferred into the slave stage (output), and the master stage is stable until Clock falls again.

DFF Timing



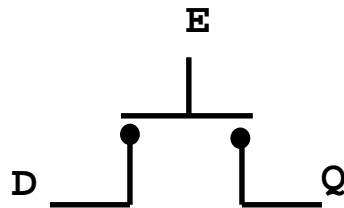
Tri-State Driver

- The Tri-State driver is like a (one directional) switch:
 - * When Enable is on ($E=1$) it transfers the input to the output.
 - * When Enable is off ($E=0$) it disconnects the output from the input.



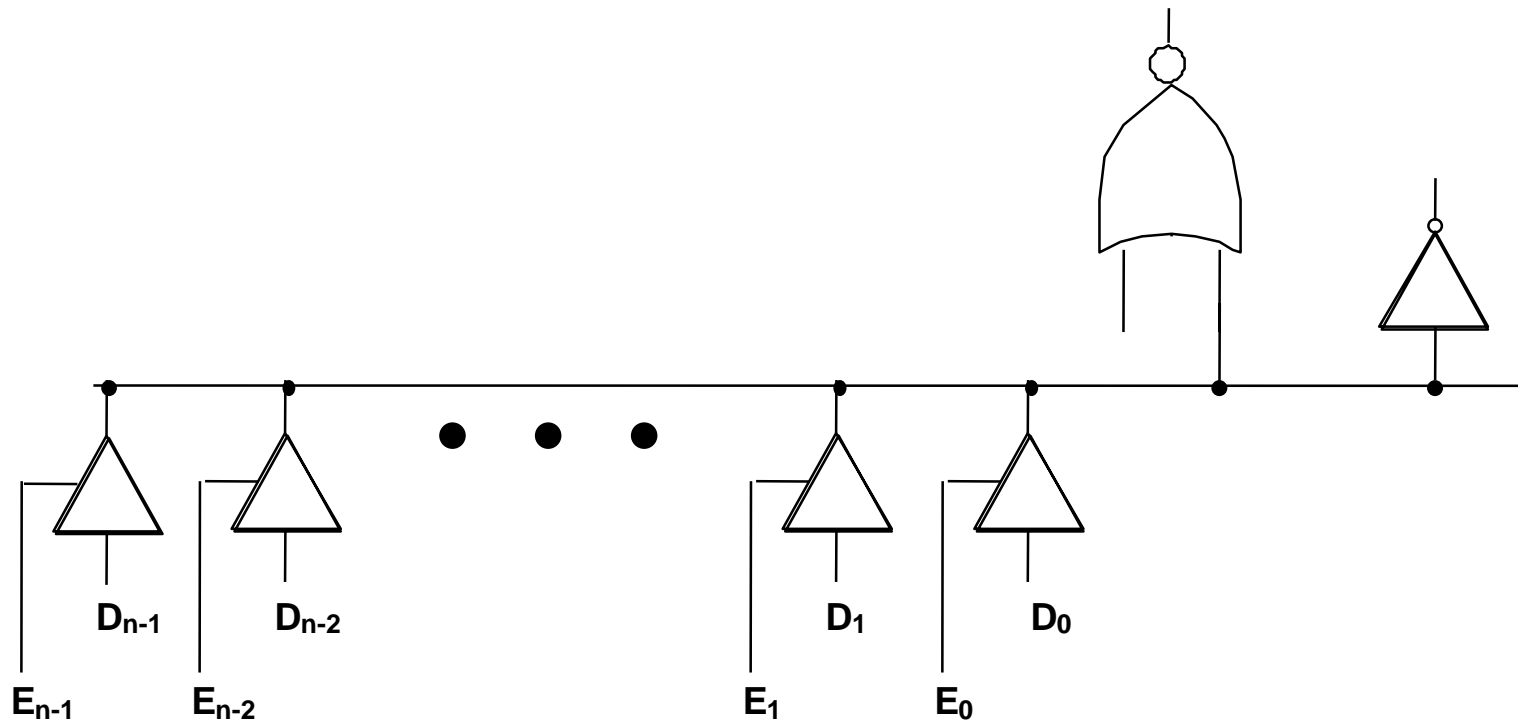
D	E	Q
0	1	0
1	1	1
-	0	Z

Z :- High Impedance

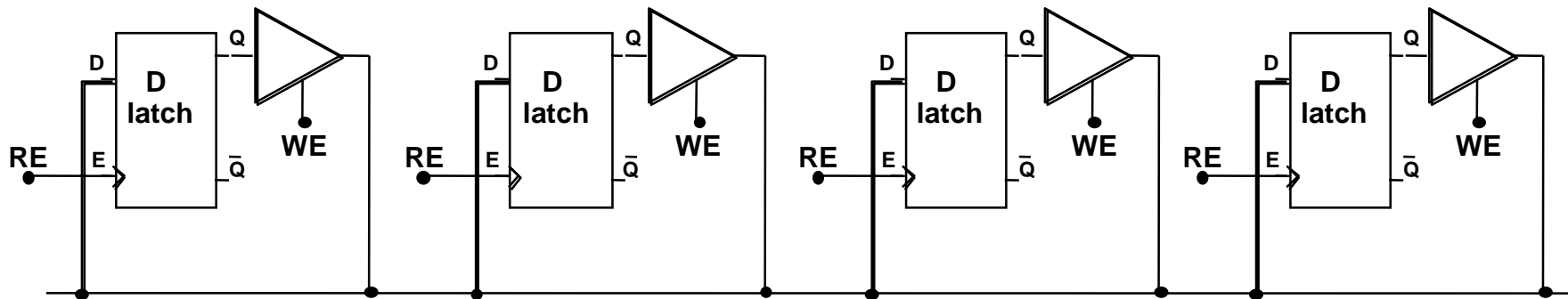


Bus Connections

- **The Bus: Many to many connections.**
- **Mutual exclusion: At most one Enable is on!**

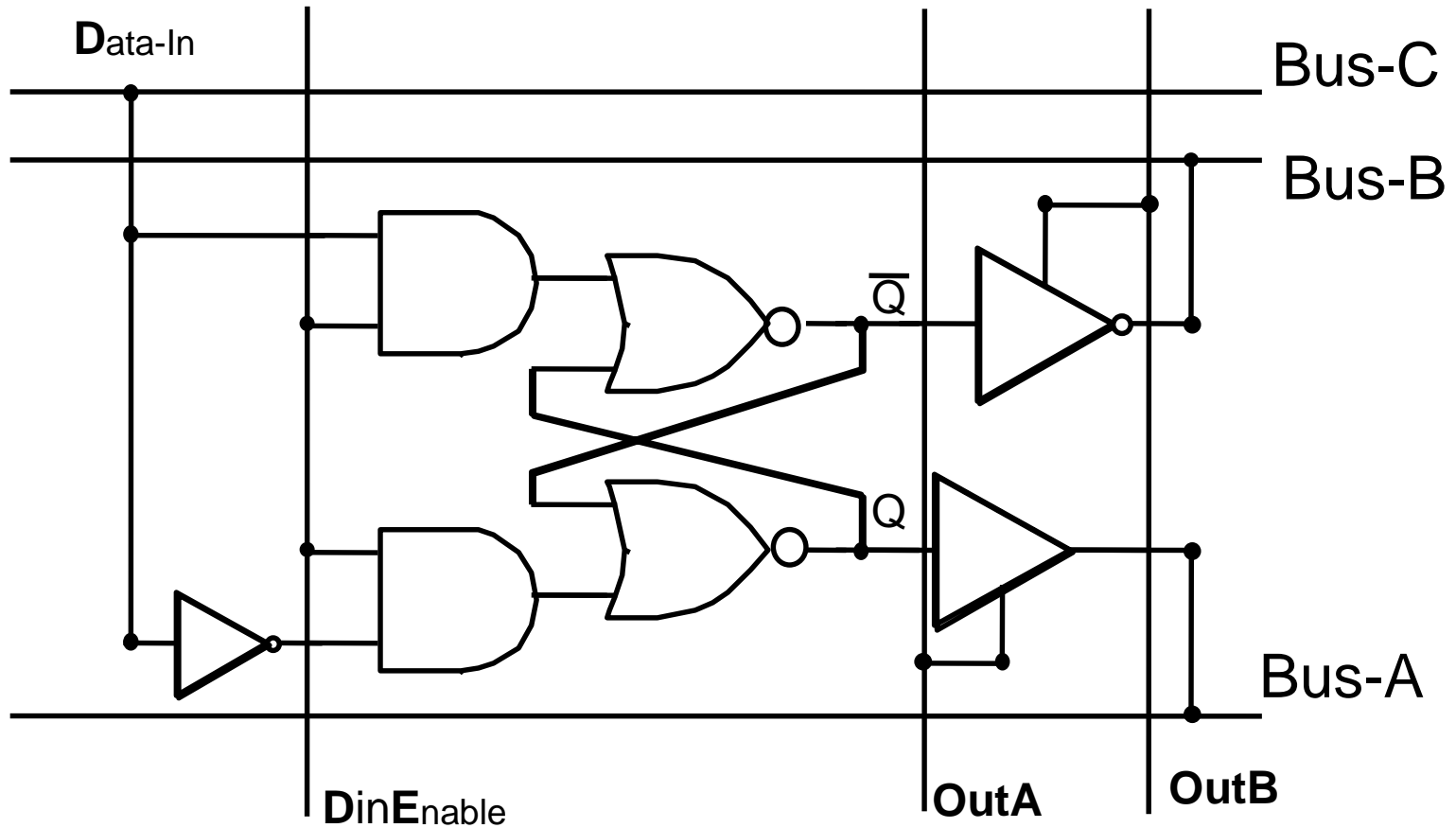


Register Cells on a bus

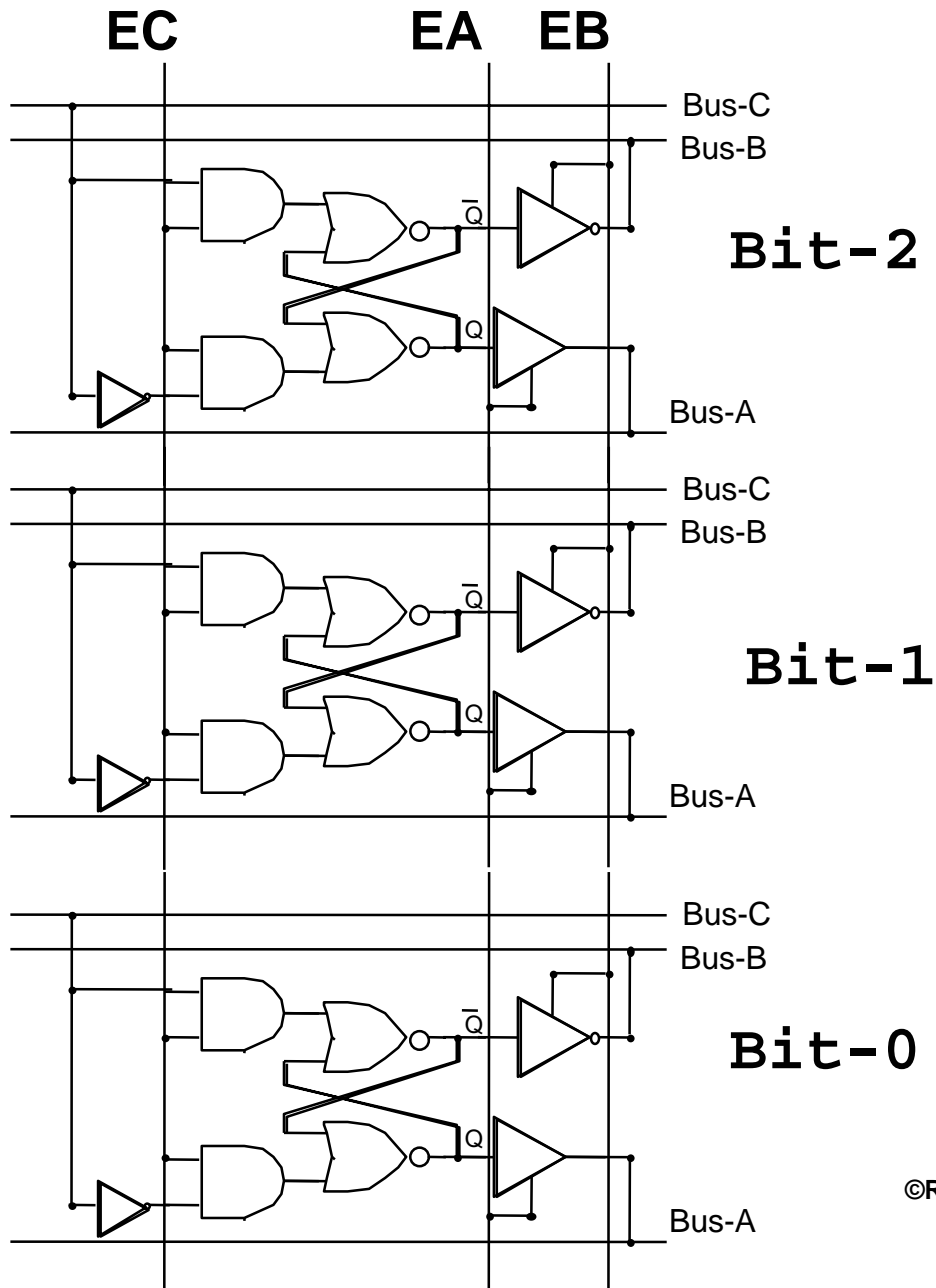


One can “source” and “sink” from any cell on the bus by activating the right controls (WE and RE).

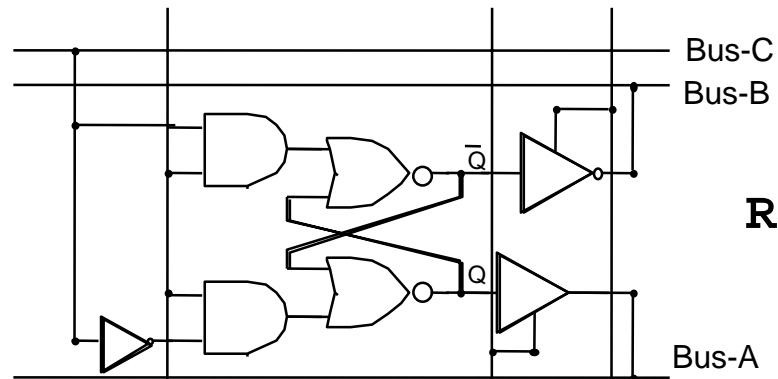
3-Port Register Cell



3-Port Register

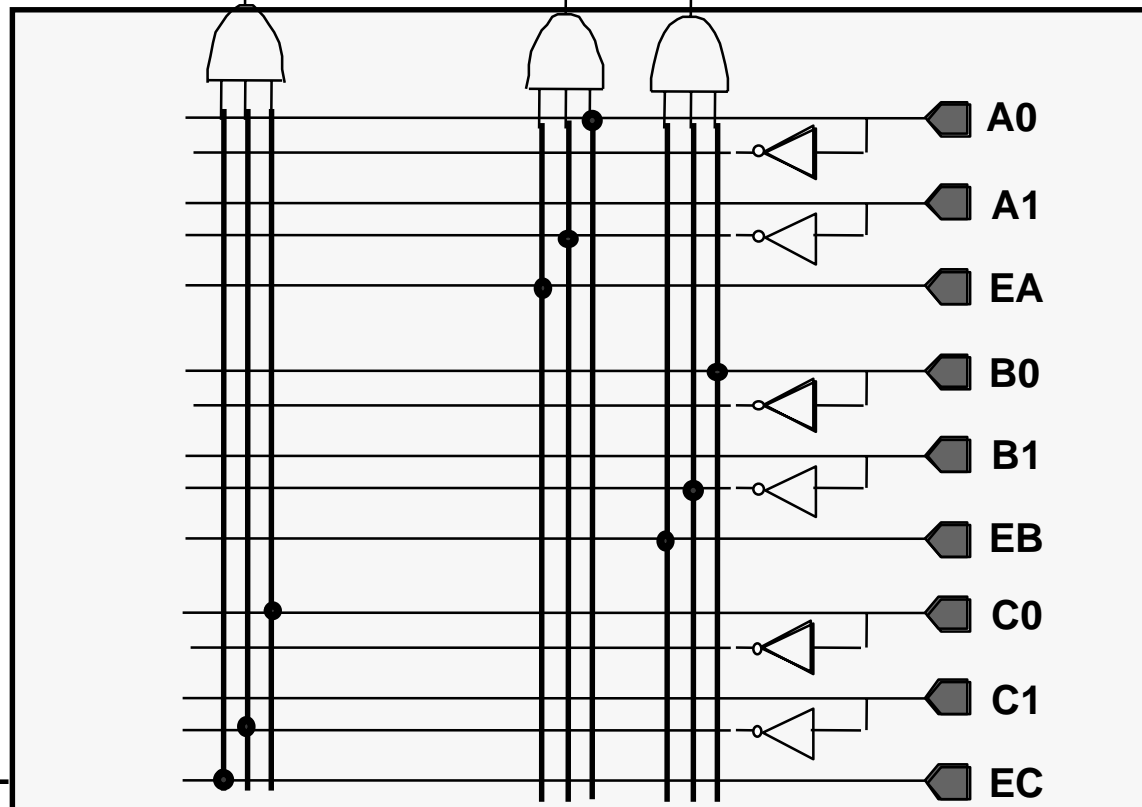


Address Decode circuit

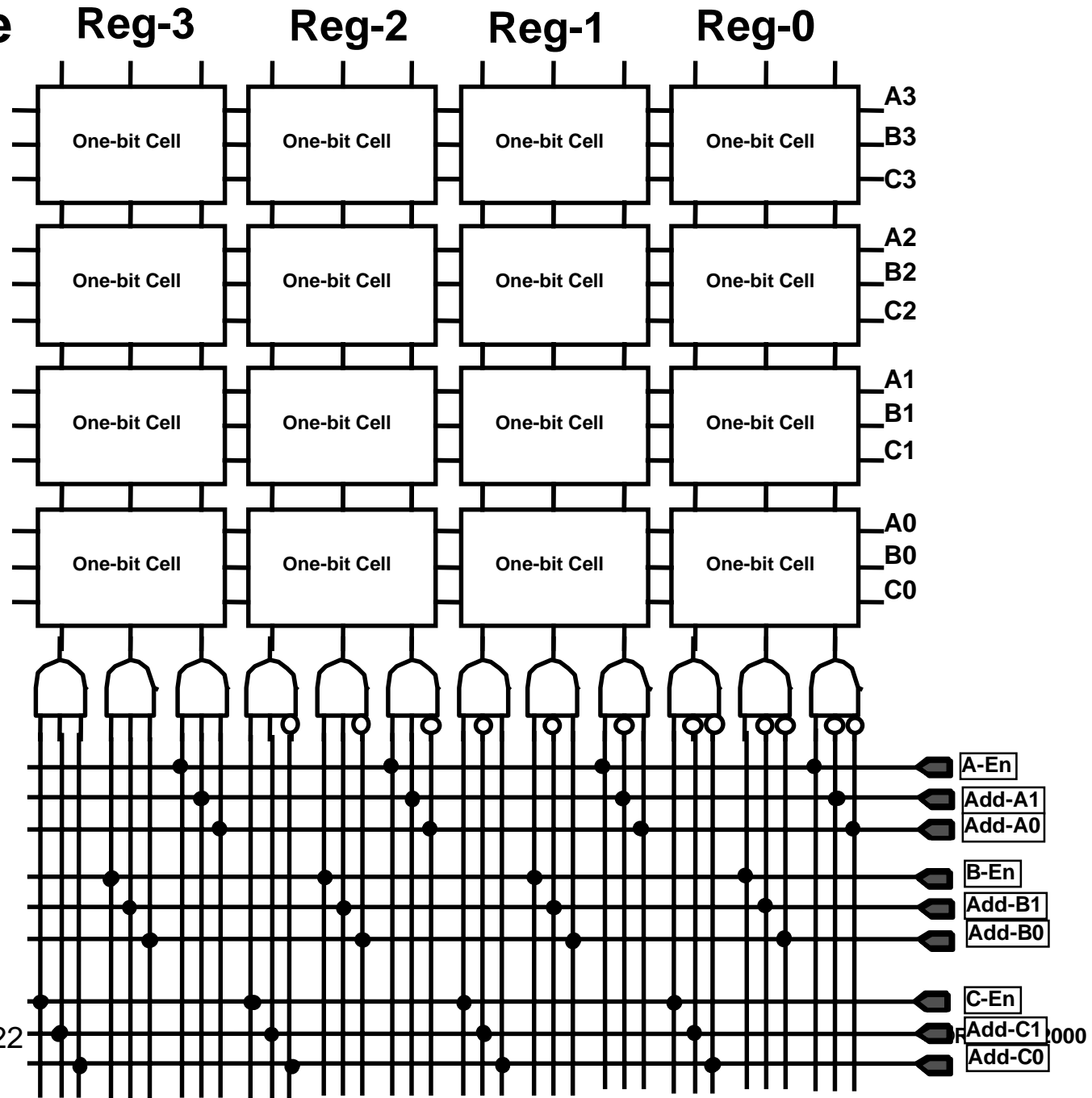


Register address: 01

For different address,
change "dots"



Register File



Addresses selected by "bubbles" on AND gates

Summary

- **So far we saw how to take a Boolean function and generate a circuit that “realizes” the function.**
- **We learned to construct circuits that can add and subtract.**
- **We learned about the ALU: a circuit that can add, subtract, detect overflow, compare, and do bit-wise operations (AND, OR, NOT)**
- **Saw how to construct a shifter circuit.**
- **Learned about the memory elements: RS-Latch, D-Latches and D-Flip-flops.**
- **Learned about Tri-State drivers and BUS Communication. (many-to many)**
- **Learned how to construct a register file.**
- **Saw how control signals can modify what the circuit will do with inputs.**
 - * **Examples: ALU, Shift, Register read-write, ...**