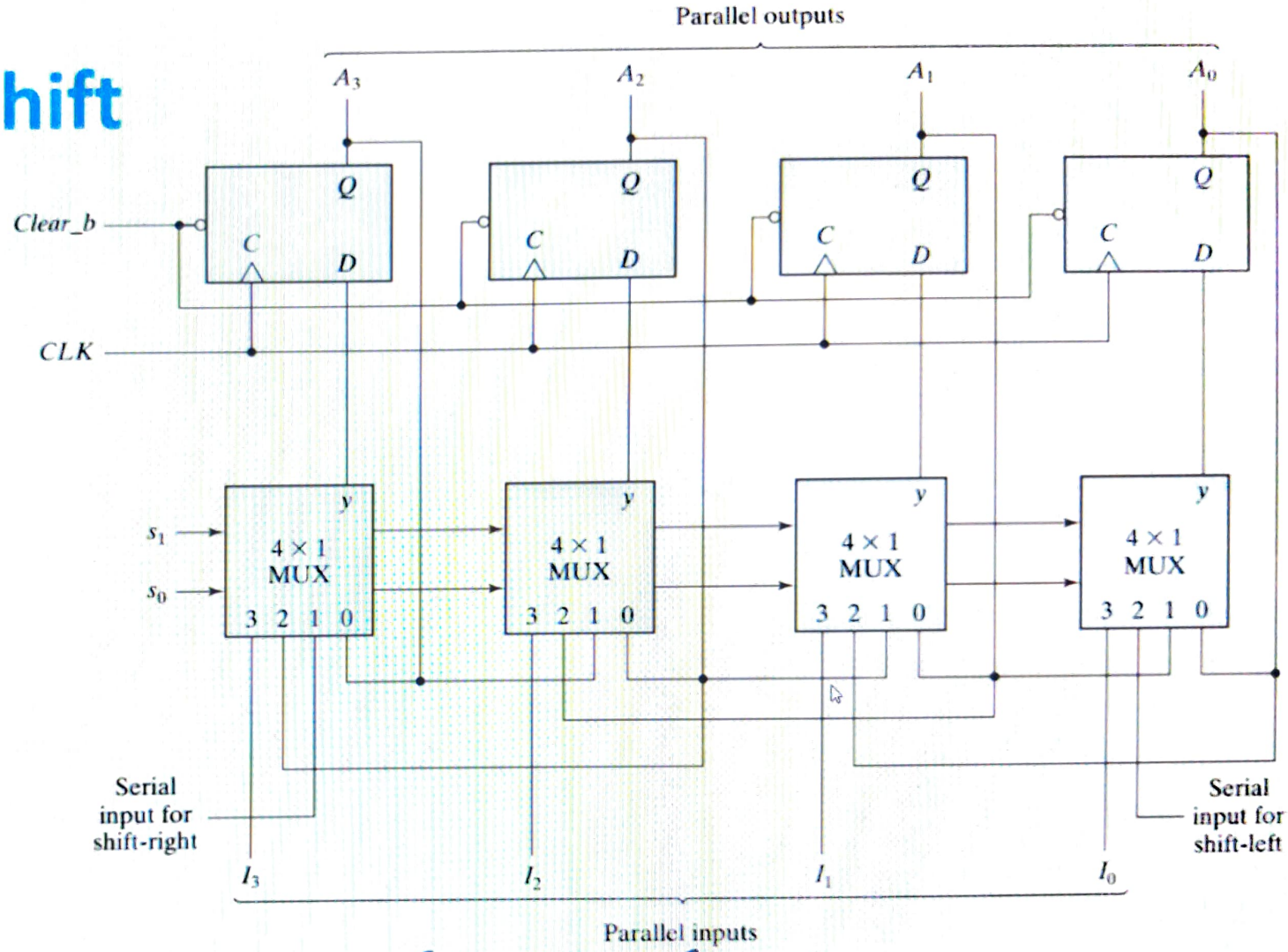
Universal Shift Register

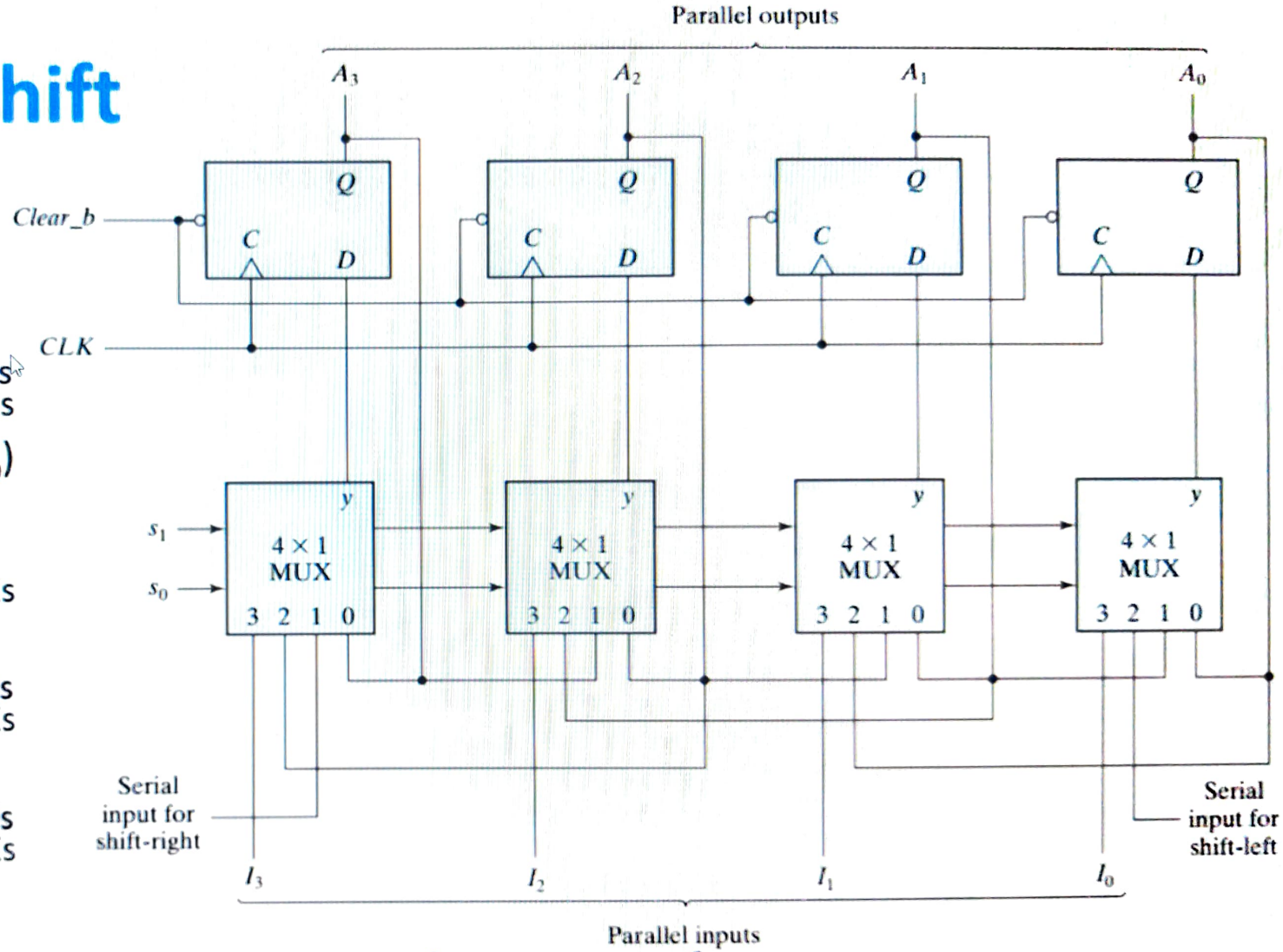
- No Shift
- Right shift
- Left shift
- Parallel load
  - S_0 S_1: 1 1→ 3rd pin is selected in all MUXs
  - Input 1 0 1 0
  - Clock pulse
  - Output 1 0 1 0



Dr. E. Paul Braineard

Universal Shift Register

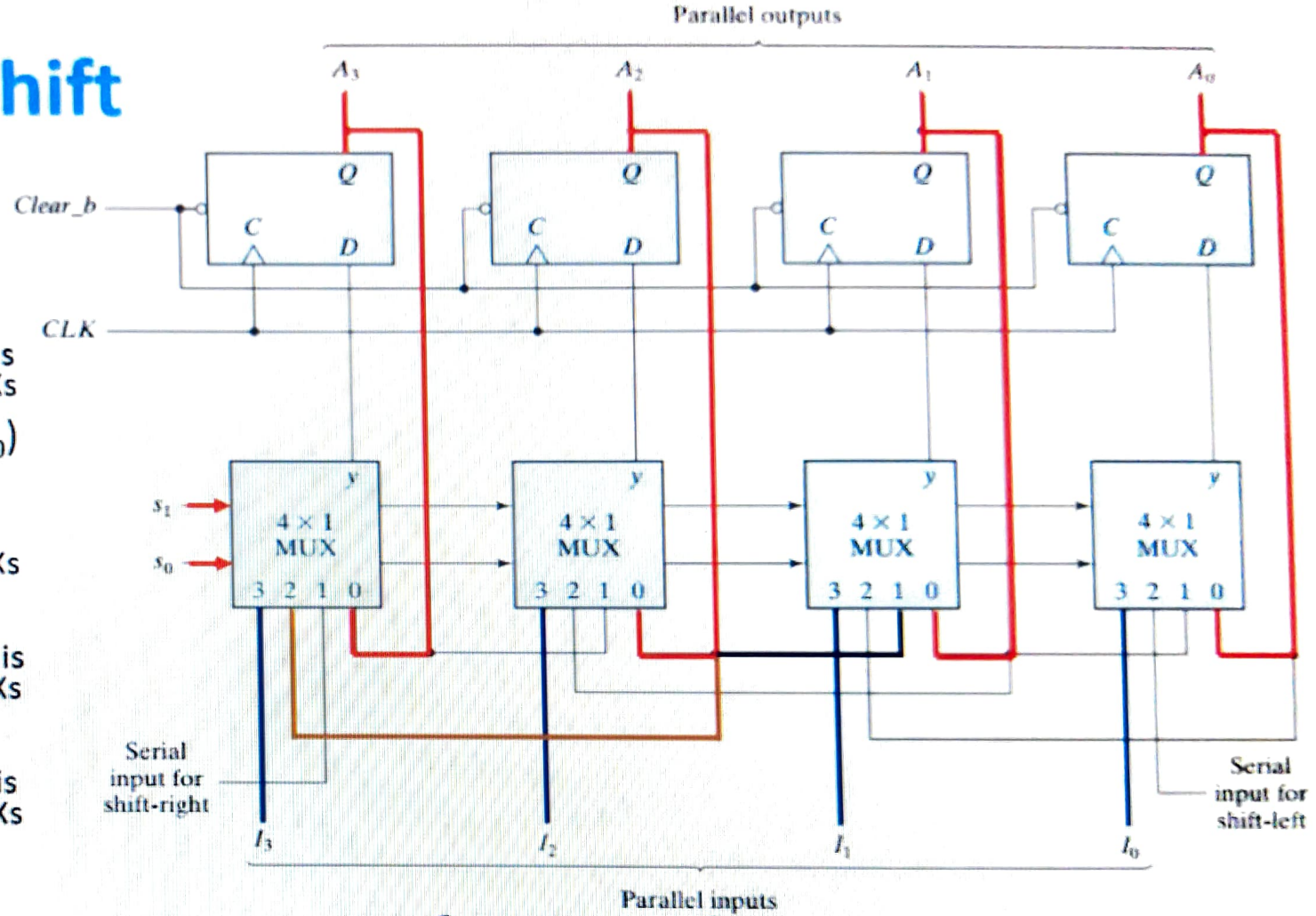
- Select lines (S_0 S_1)
- Input lines ($I_3 - I_0$)
  - S_0 S_1: 1 1→ 3rd pin is selected in all MUXs
- Output lines ($A_3 - A_0$)
- No Shift
  - S_0 S_1: 0 0→ 0 pin is selected in all MUXs
- Right shift
  - S_0 S_1: 1 0→ 3rd pin is selected in all MUXs
- Left shift
  - S_0 S_1: 0 1→ 3rd pin is selected in all MUXs



Dr. E. Paul Braineard

Universal Shift Register

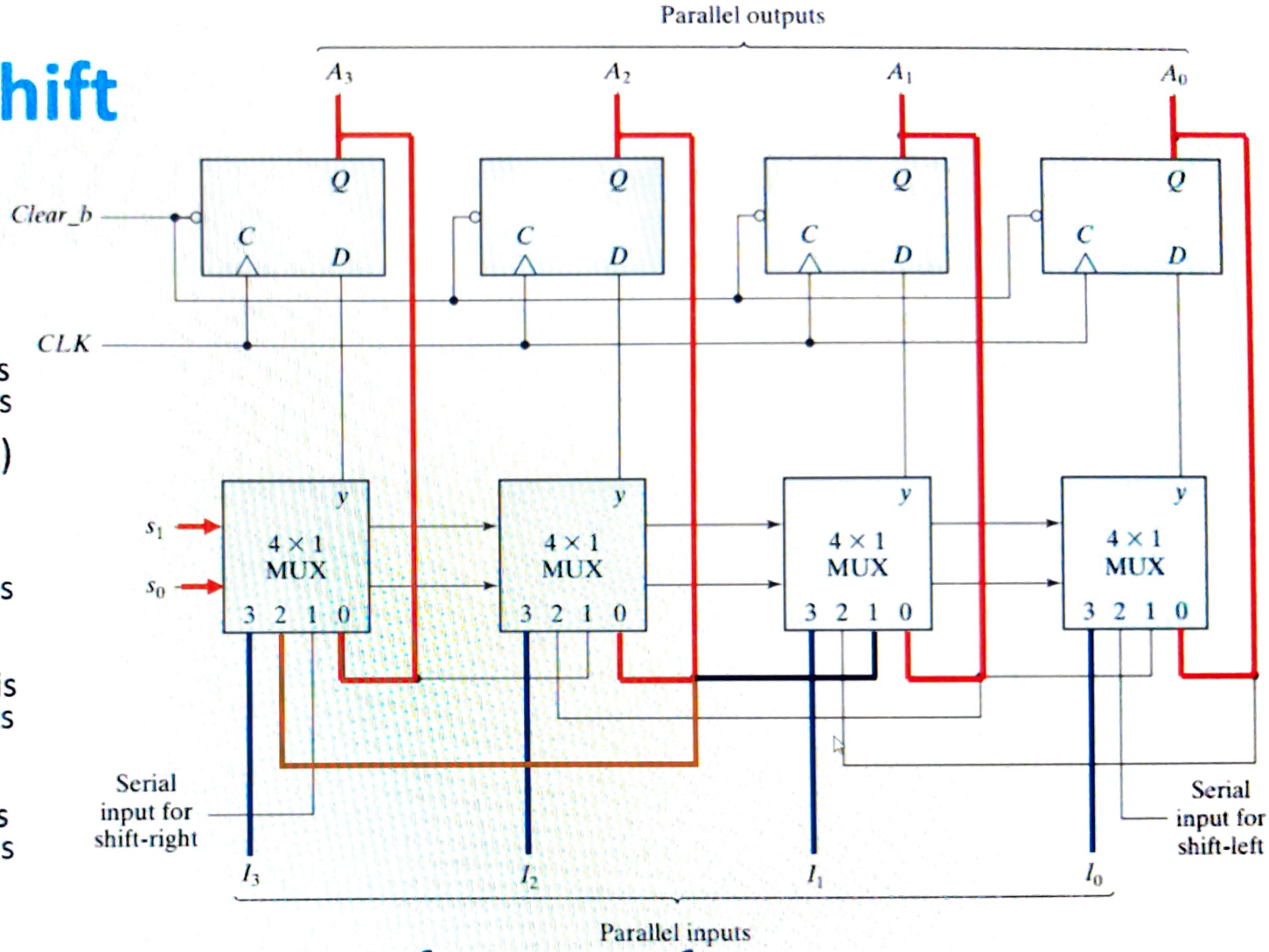
- Select lines (S_0 S_1)
- Input lines ($I_3 - I_0$)
  - S_0 S_1: 1 1→ 3rd pin is selected in all MUXs
- Output lines ($A_3 - A_0$)
- No Shift
  - S_0 S_1: 0 0→ 0 pin is selected in all MUXs
- Left shift
  - S_0 S_1: 1 0→ 2nd pin is selected in all MUXs
- Right shift
  - S_0 S_1: 0 1→ 1st pin is selected in all MUXs



Dr. E. Paul Braineard

Universal Shift Register

- Select lines (S_0 S_1)
- Input lines ($I_3 - I_0$)
  - S_0 S_1: 1 1→ 3rd pin is selected in all MUXs
- Output lines ($A_3 - A_0$)
- No Shift
  - S_0 S_1: 0 0→ 0 pin is selected in all MUXs
- Left shift
  - S_0 S_1: 1 0→ 2nd pin is selected in all MUXs
- Right shift
  - S_0 S_1: 0 1→ 1st pin is selected in all MUXs

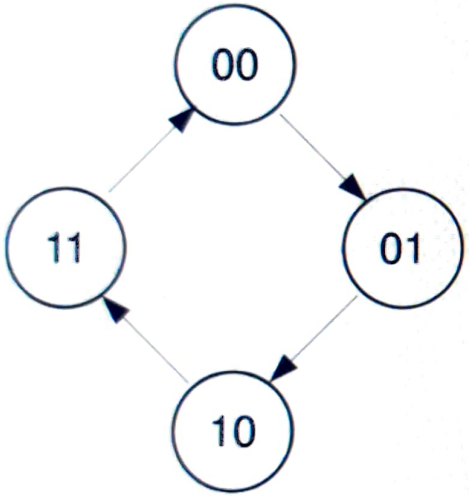


Dr. E. Paul Braineard

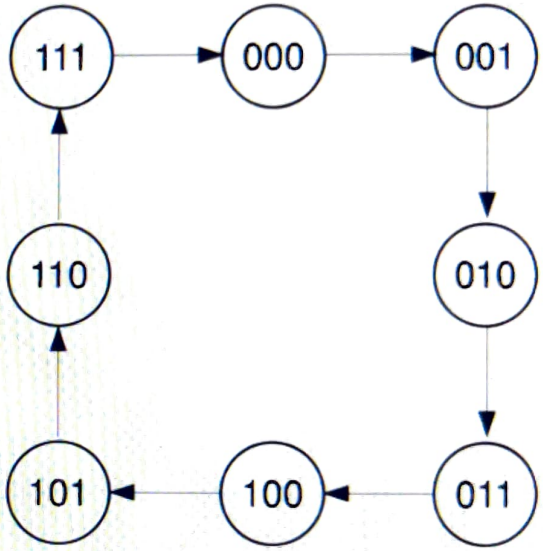
Counters

- Sequential circuits that count through a specific sequence of states
  - Count up
  - Count down
  - Count through other fixed sequences
- State: The state of counter is stored in FFs
- n-bit counter
  - Has 'n' FFs
  - Can cycle (Count) through 2^n states

Counters



Two bit counter
(Four states)



Three bit counter
(Eight states)

Dr. E. Paul Braineard

Counters: Examples

- Binary counter: 000, 001, 010, 011, 100, 101, 110, 111, 000
- Gray code counter:
  - 000, 010, 110, 100, 101, 111, 011, 001, 000, 010, 110
- One-hot counter: 0001, 0010, 0100, 1000, 0001, 0010, ...
- BCD counter: 0000, 0001, 0010, ..., 1001, 0000, 0001
- pseudo-random sequence generators: 10, 01, 00, 11, 10, 01, 00, ...

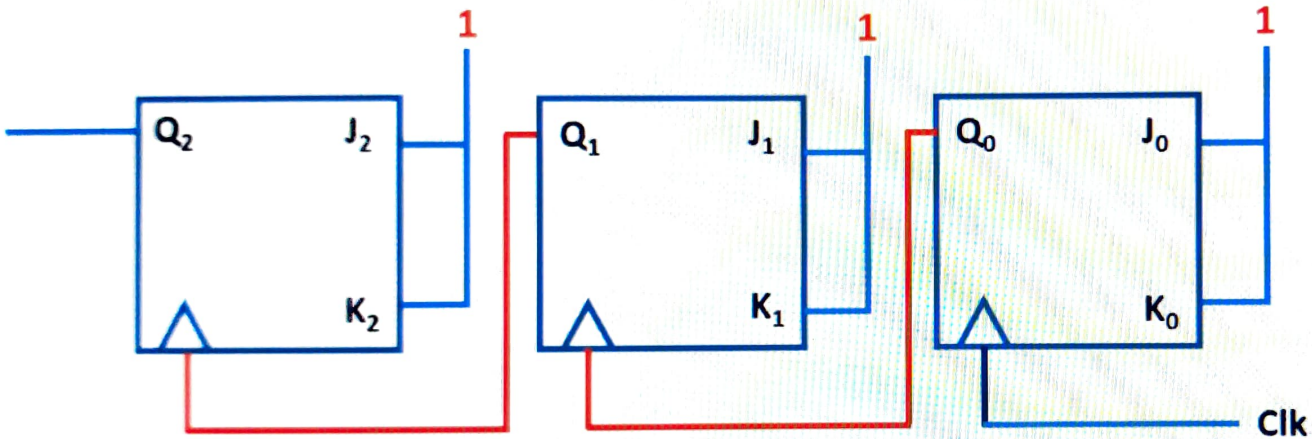
Types of Counters

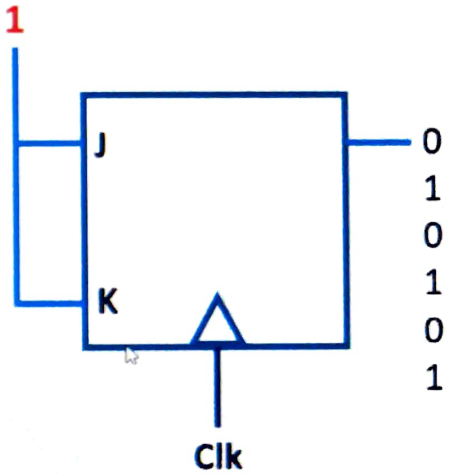
- Ripple counter
  - Clock connected on the LSB bit FF
  - For all other bits, a FF output is connected to the clock input of other FF starting from LSB FF (Asynchronous)
  - Output change is delayed by one clock period towards MSB
- Synchronous counter
  - Clock is directly connected to all flip-flop clock inputs
  - Logic is used to implement the desired state sequencing

Counters: Design

1. Draw a state graph
   - It specifies the desired sequence of the counter
2. Construct a state table (from the state graph)
   - One Flip-Flop for each bit in the state
3. Derive a K-map (from the state table for each Flip-Flop input)
   - Select the type of Flip-Flop to be used
4. Determine the input equation(s) for each Flip-Flop

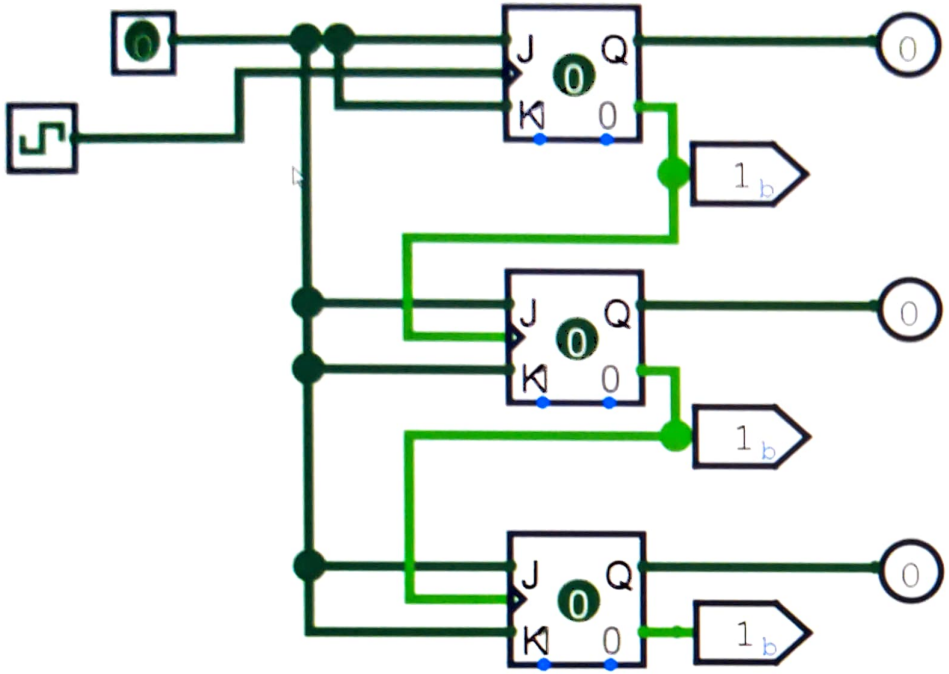
3-bit Asynchronous counter





Dr. E. Paul Braineard

3-bit Asynchronous Up counter

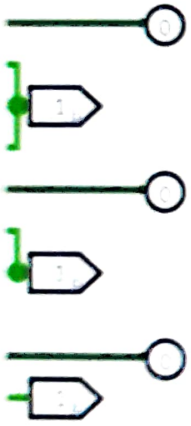


Rising edge

Clk_2	Clk_1	Clk_0	Q_2	Q_1	Q_0
Reset			0	0	0
11	10	01	0	0	1
11	00	10	0	0	1

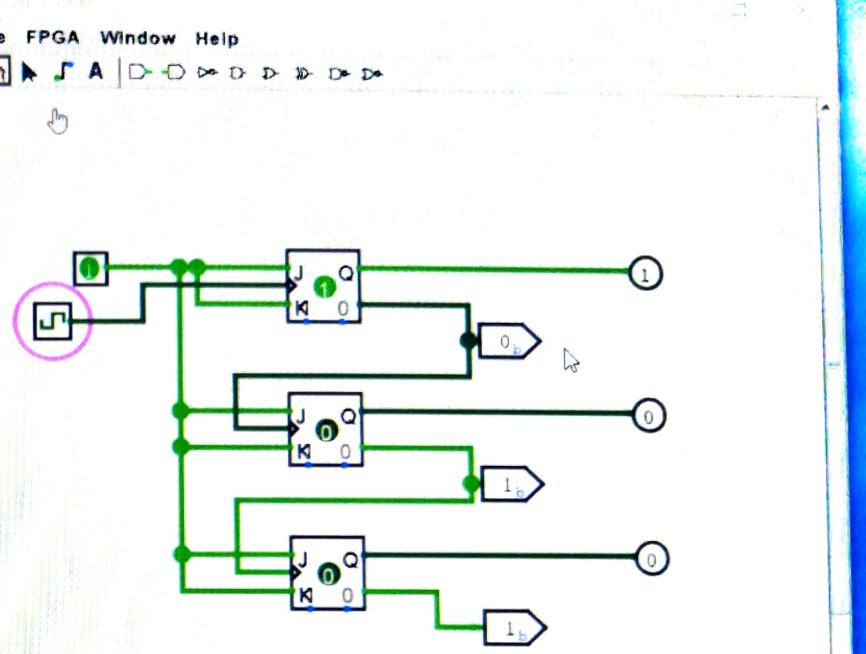
Dr. E. Paul Braineard

s Up counter



Clk_2	Clk_1	Clk_0	Q_2	Q_1	Q_0
Reset			0	0	0
11	10	01	0	0	1
11	00	10	0	0	1
10	01	01	0	1	0
00	10	01	0	1	1
01	01	01	1	0	0
11	10	01	1	0	1
10	01	01	1	1	0
00	10	01	1	1	1

S. Paul Braineard



Raise hand

Turn on captions