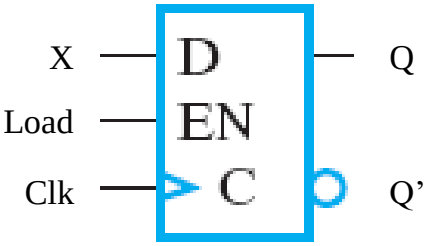
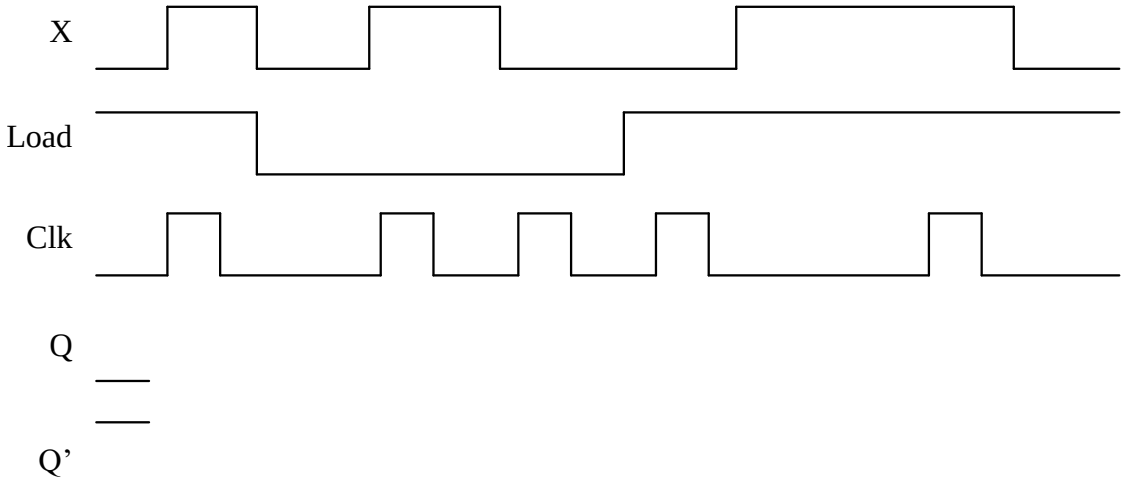
Tutorial 7 – Sample Solution
Registers and RTL

CO 2206 Computer Organization

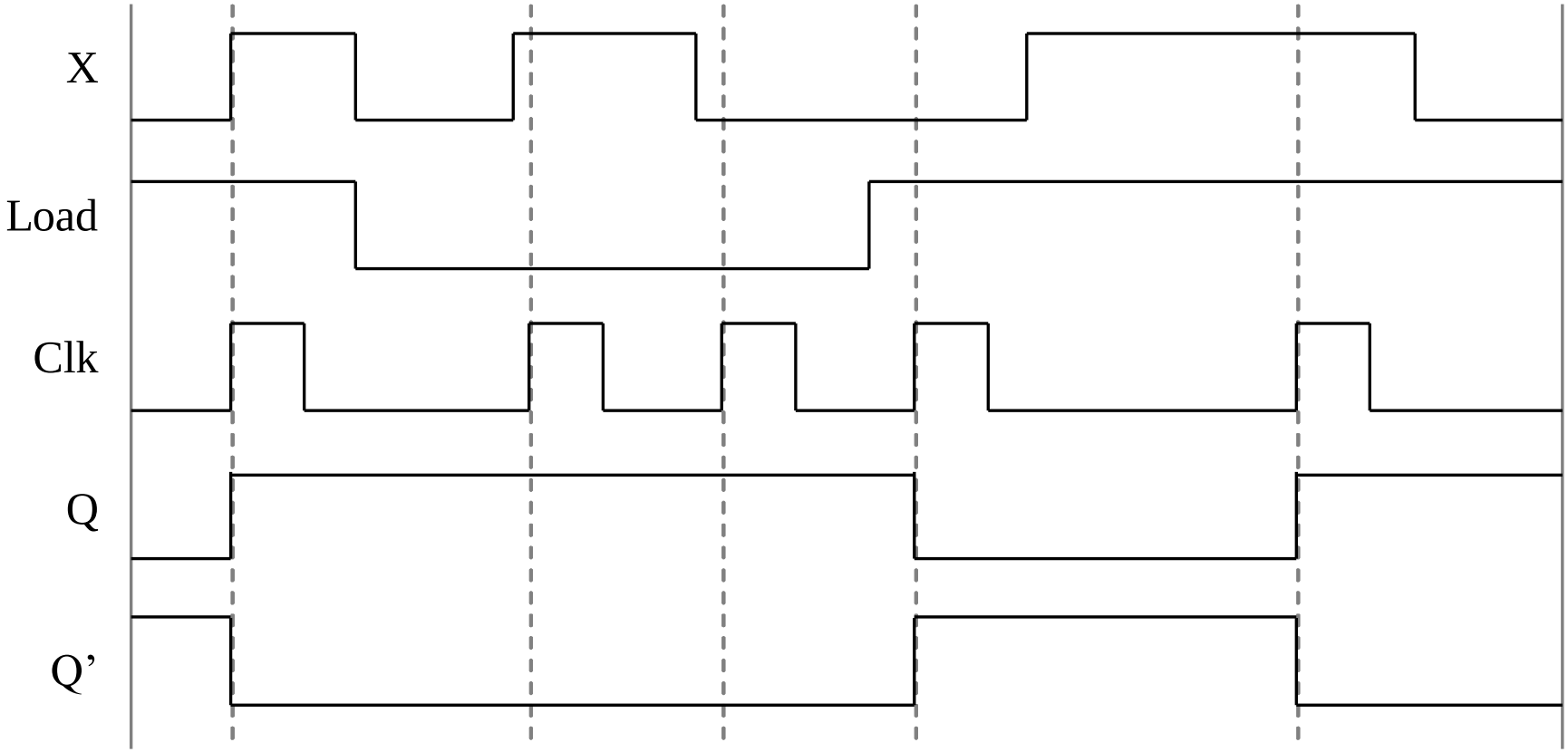
Task 1. Register

- Knowing the function of a D flip-flop with enable, complete the following timing diagram (for the outputs). Assume initially Q=0 and Q'=1.





Task 1. Ans

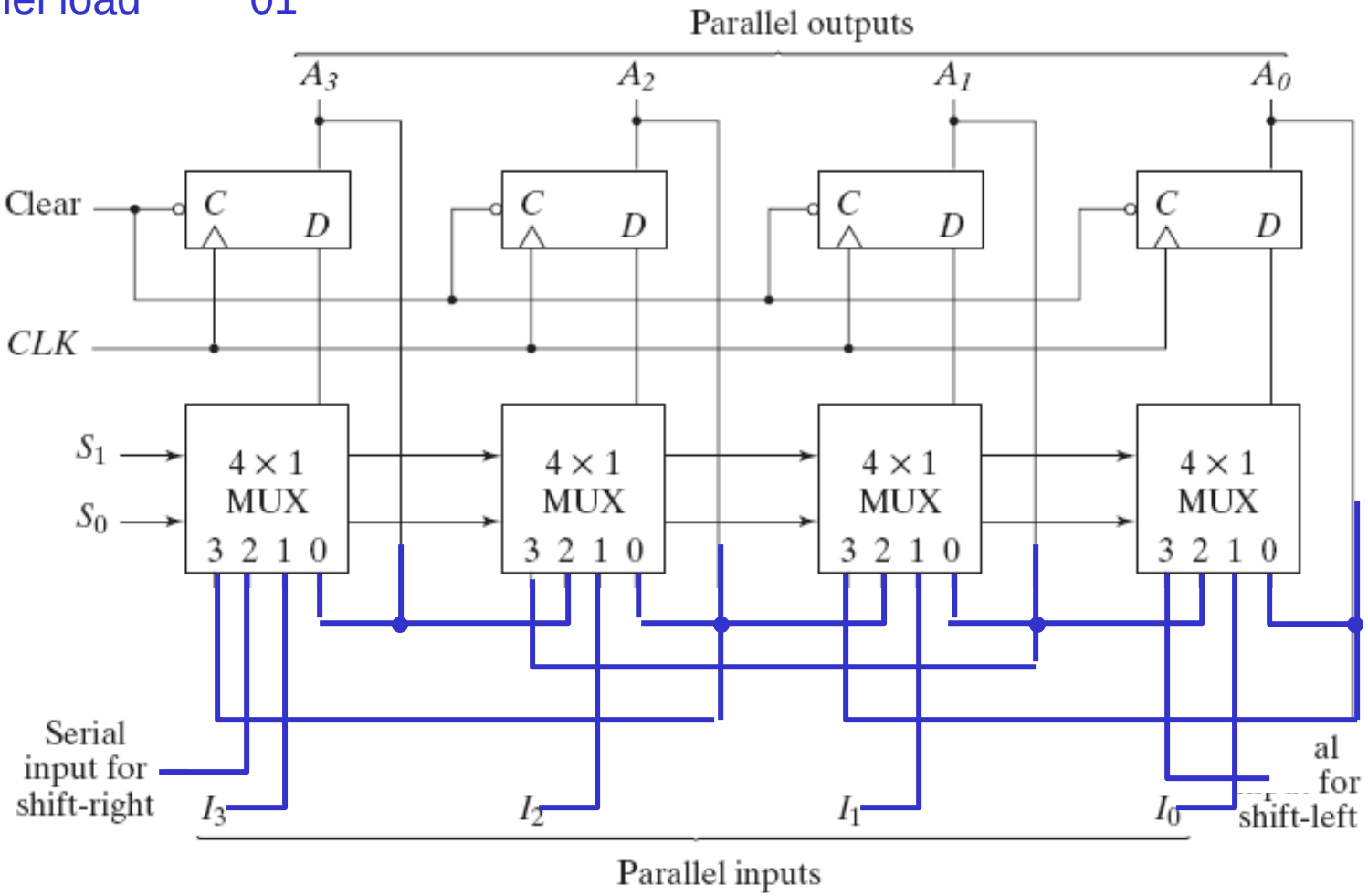


Task 2. Shift Register

- Modify the 4-bit universal shift register on Slide 19, Chapter 7, such that the functions are:
  - when s1s0 = 00,
    - no change of states
  - when s1s0 = 01,
    - parallel load
  - when s1s0 = 10,
    - shift right
  - when s1s0 = 11,
    - shift left

Lecture Slide:	Tutorial
00 – no change	00
01 – shift right	10
10 – shift left	11
11 – parallel load	01

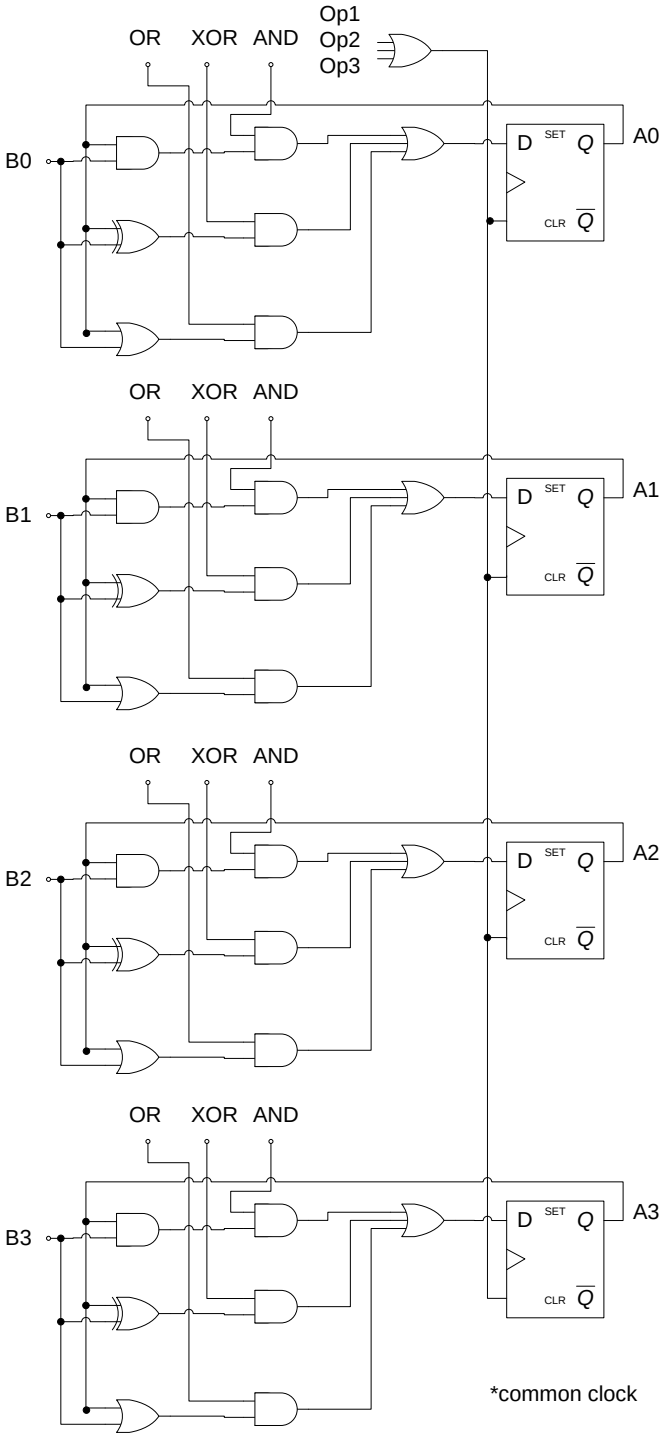
Task 2. Ans



Task 3. Register Cell Design

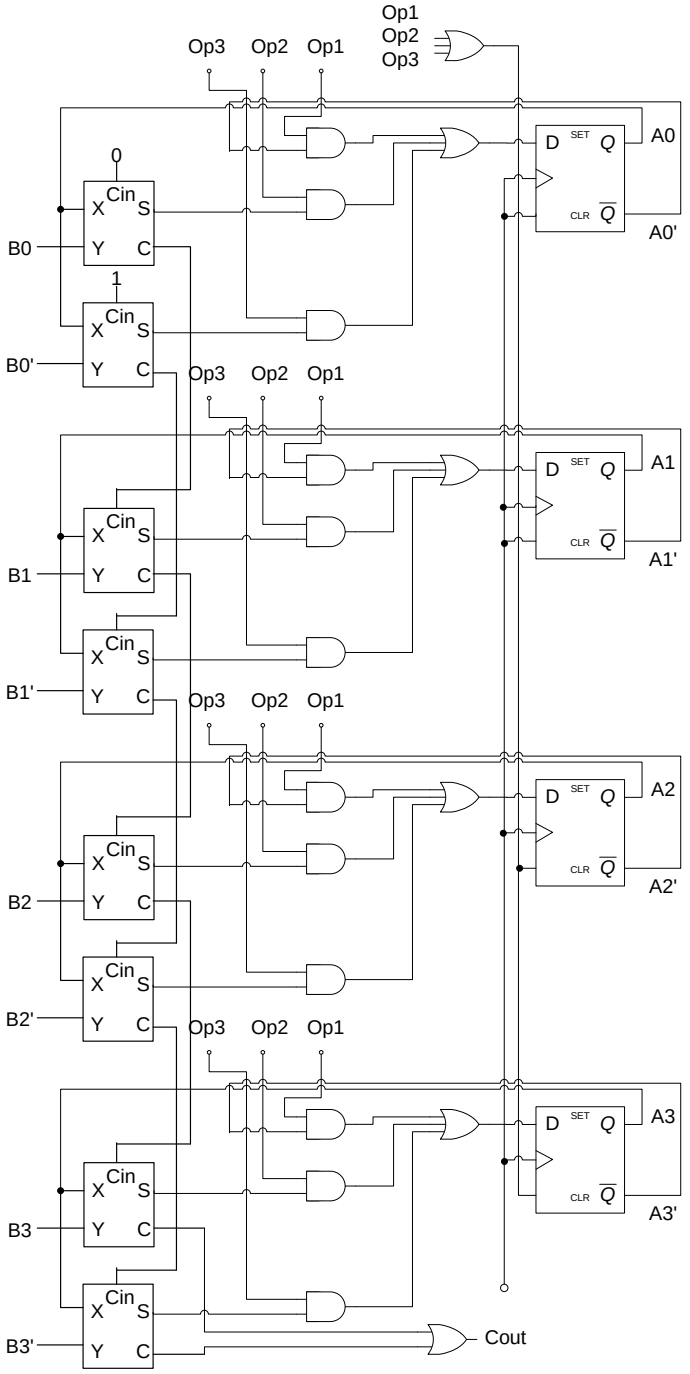
- Draw the circuit diagram to implement the ad-hoc register cell design on Slide 41, Chapter 7
- By referring to the ad-hoc design above, design (and draw the circuit of) a register cell to implement the following register transfer:
  - Op1: A ← A'
  - Op2: A ← A + B
  - Op3: A ← A – B
  - Assume that
    - Only one of Op1, Op2, Op3 is equal to 1
    - For Op1, Op2, Op3 equal to 0, A remains unchanged
    - You can use NOT gate(s) and a Full Adder in your design.

Task 3. Ans (1)



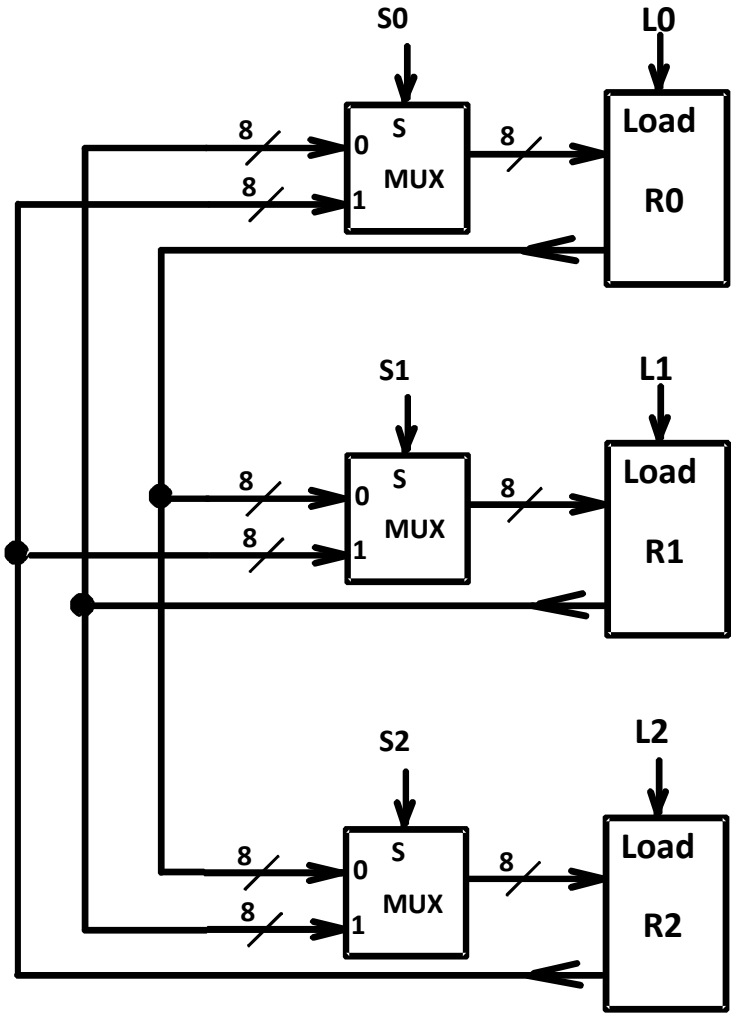
*common clock

Task 3. Ans (2)


Op1
Op2
Op3
Op3
Op2
Op1
0
1
Cin
X
Y
S
C
B0
B0'
B1
B1'
B2
B2'
B3
B3'
D
SET
Q
CLR
A0
A0'
A1
A1'
A2
A2'
A3
A3'
Cout

Task 4. Mux-Based Transfer

- Determine the value for the select (S2, S1, S0) signals and load (L2, L1, L0) signals to perform the following register transfer operations:
  - **R0 ← R1**
  - **R1 ← R2**
  - **R1 ← R0, R2 ← R0**



Task 4. Ans

- **R0 ← R1**
  - **L2=0,L1=0,L0=1,S2=0,S1=0,S0=0**
- **R1 ← R2**
  - **L2=0,L1=1,L0=0,S2=0,S1=1,S0=0**
- **R0 ← R1, R0 ← R2**
  - **L2=1,L1=1,L0=0,S2=0,S1=0,S0=0**

Task 5. RTL

- Express the following statement in *RTL*

```
if (ABC = 001) then
(R0 = R1) else if
(ABC = 010) then (R0
= R2) else if (ABC =
011) then (R0 = RR3)
else if (ABC = 101)
then (R0 = R0 OR R1)
```

Operation	Text RTL
Combinational Assignment	=
Register Transfer	←
Addition	+
Subtraction	–
Bitwise AND	^
Bitwise OR	∨
Bitwise XOR	⊕
Bitwise NOT	—
Shift left (logical)	sl
Shift right (logical)	sr
Vectors/Registers	A(3:0)
Concatenation	||

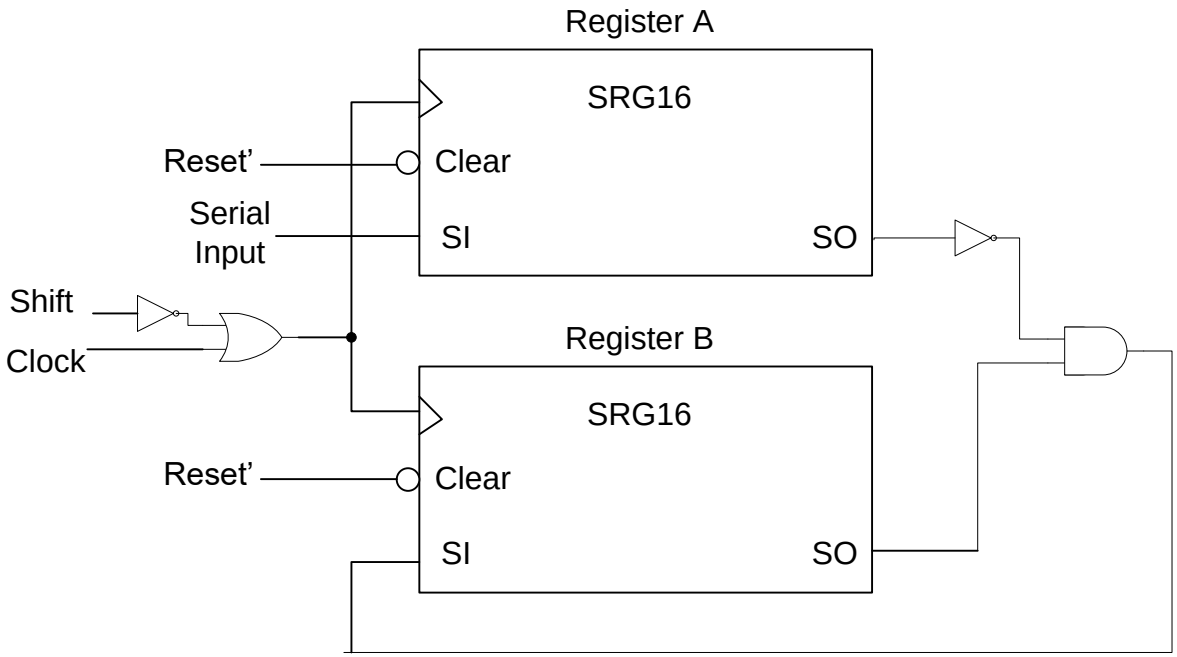
Task 5. Ans

- **if (ABC = 001) then (R0 = R1) else**
  if (ABC = 010) then (R0 = R2) else
  if (ABC = 011) then (R0 = R3) else
  if (ABC = 101) then (R0 = R0 OR R1)
- **A′B′C: R0 ← R1, A′BC′: R0 ← R2,**
  A′BC: R0 ← R3, AB′C: R0 ← R0 V R1

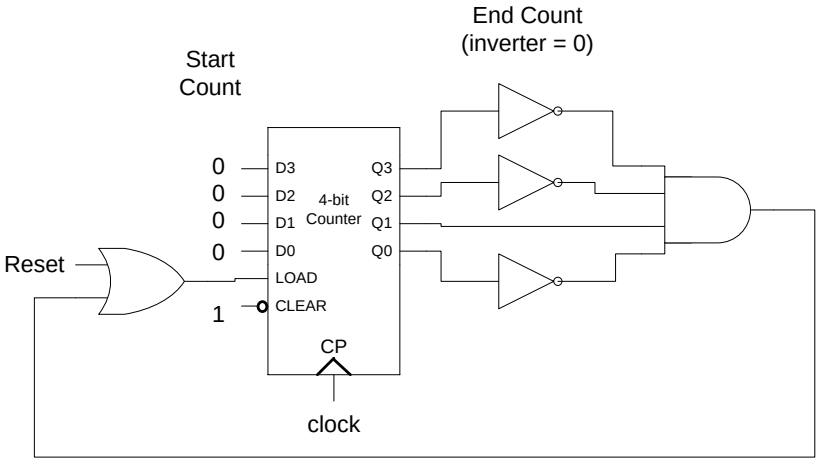
Task 6. Serial Register Operation

- Modify the circuit on Slide 59 of Chapter 7 to perform a serial operation of the following logic function:
  - **`B = A′B where A and B are both 16-bit`**

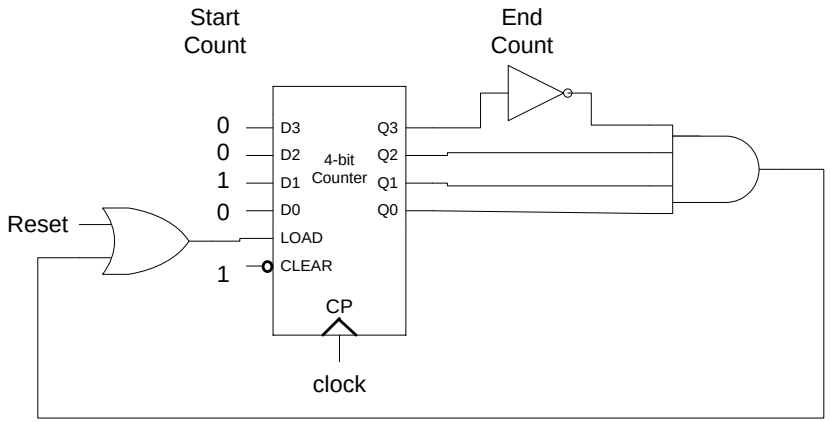
Task 6. Ans


Register A
SRG16
Reset'
Clear
Serial
Input
SI
SO
Shift
Clock
Register B
SRG16
Reset'
Clear
SI
SO

Task 7: Ans.


Start
Count
End Count
(inverter = 0)
0
0
0
0
1
D3
D2
D1
D0
LOAD
CLEAR
4-bit
Counter
Q3
Q2
Q1
Q0
CP
clock
Reset

a.


Start
Count
End
Count
0
0
1
0
1
D3
D2
D1
D0
LOAD
CLEAR
4-bit
Counter
Q3
Q2
Q1
Q0
CP
clock
Reset

b.