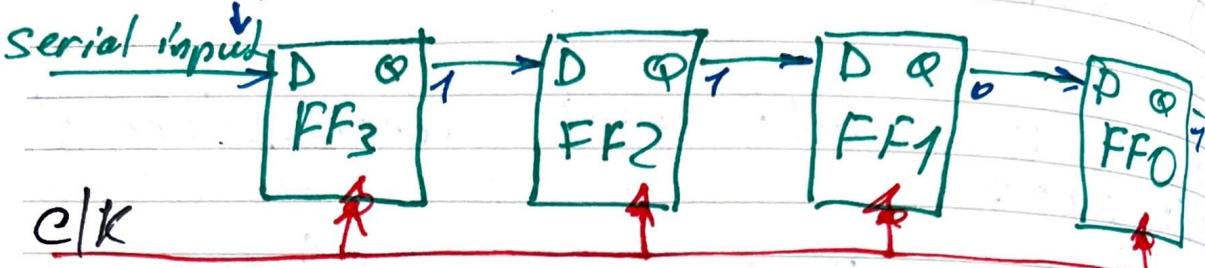


① The content of a 4-bit is initially 1101. The register is shifted six times to the right with the serial input. What is the content of the register after each shift?

101101<sub>2</sub> LSB

SISO



clk	$Q_3$	$Q_2$	$Q_1$	$Q_0$
-----	-------	-------	-------	-------

Initial State	1	1	0	1
---------------	---	---	---	---

	1	1	0	1
--	---	---	---	---

	0	1	1	1
--	---	---	---	---

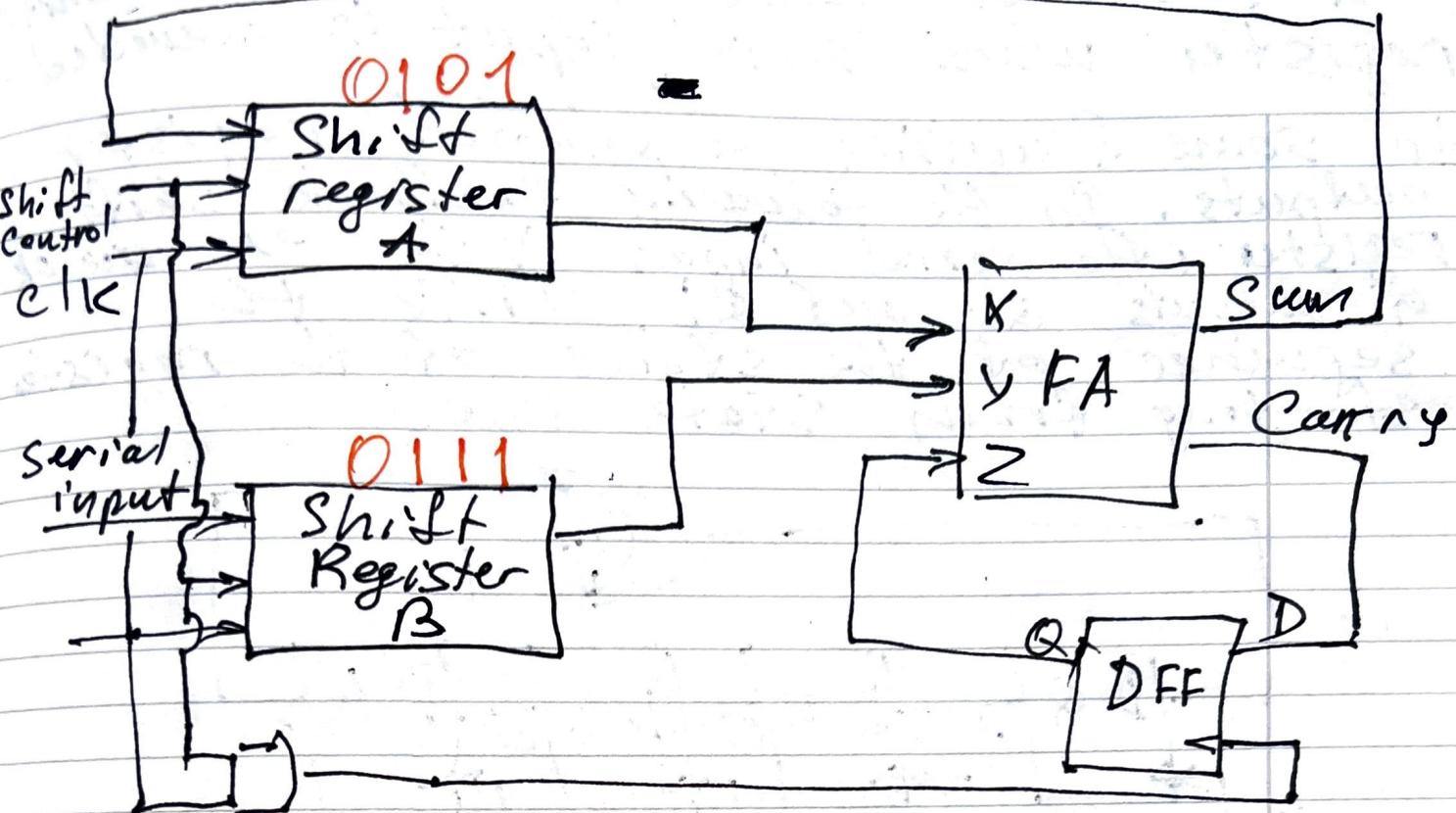
	1	0	1	1
--	---	---	---	---

	1	1	0	1
--	---	---	---	---

	0	1	1	0
--	---	---	---	---

	1	0	1	1
--	---	---	---	---

② The serial adder uses two 4-bit registers. Register A holds the binary number 0101 and register B holds the carry flip-flop. List the binary values in register B and the carry flip-flop after each shift.



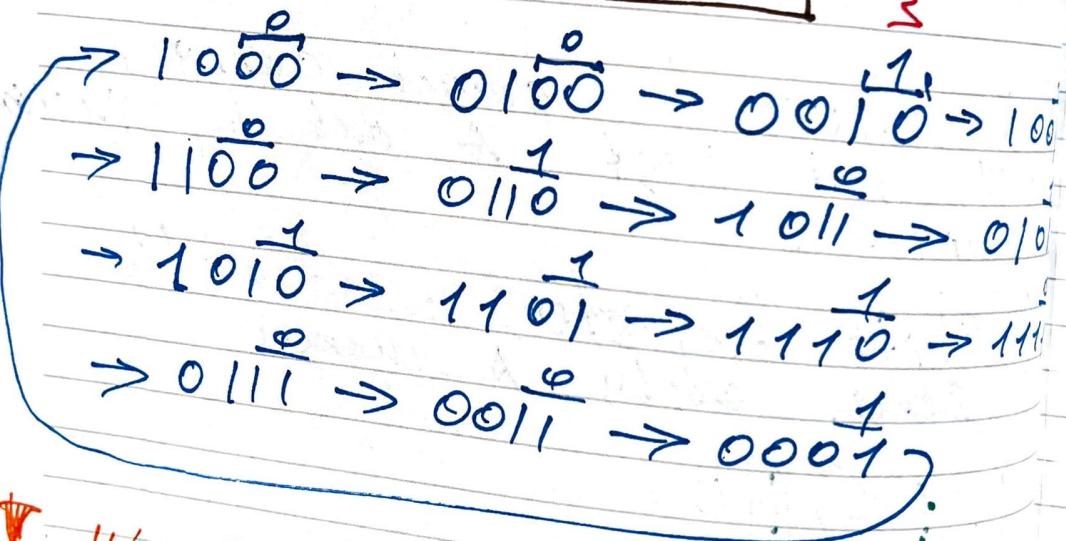
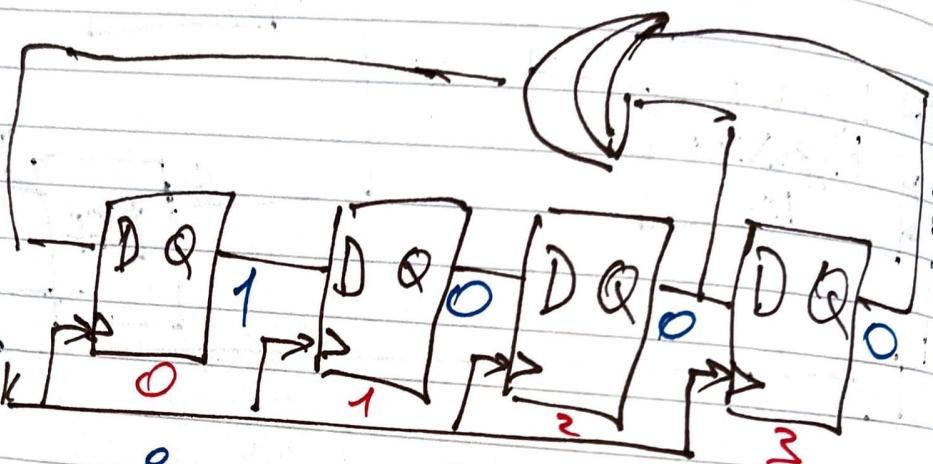
!! We have to count carry as well, not just A and B, initial carry = 0.

!! ~~Shift Register B stage the sum, only A changes~~

$$\begin{array}{l}
 \text{A} \quad 1 \quad 0 \quad 0 \quad 0 \\
 \text{B} \quad 1 \quad 1 \quad 1 \quad 0 \\
 \text{C} \quad 0 \quad 0 \quad 1 \quad 1 \\
 \text{---} \\
 \text{= } 00\cancel{1}1 \\
 \text{= } 1100
 \end{array}$$

!! We locate it from right not left

③ A feedback shift register whose serial input is to some function of selected outputs. In the following feedback register, the serial input is the function of outputs  $Q_3$  and  $Q_2$ . Write the sequence of the states of the starting from state 1000.

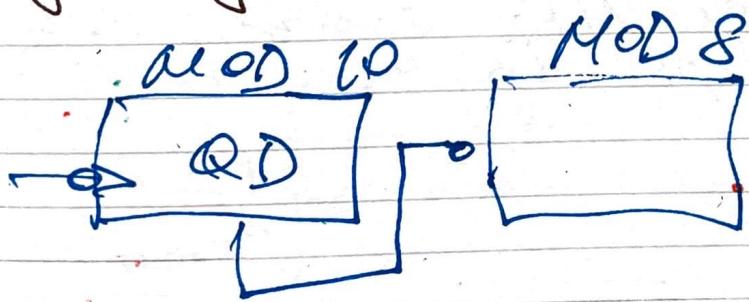


! We keep doing it until we get to original sequence

(A) The MOD number of a counter is equal to the number of complete states that a counter goes through before it recycles back to its start up state. To construct a ripple counter with a MOD number  $X$ , it requires a minimum number of  $n$  FF such that  $2^n \geq X$ , what is the maximum MOD # of a counter constructed with 6 flip flops?

~~from 0 to~~  
 Counter counts on equation of  $2^{n-1}$  where  $n$  is amount of flip flops, therefore  $2^6 = 64 - 1 = 63$ .

b) Counters with MOD number larger than 16 can be created by cascading 4-bit binary counters. The MOD number is equal to the product of the individual MOD numbers. For example, a MOD 80 counter can be implemented as shown in the following figure.



What's the largest MOD counter that can be implemented by cascading two 4-bit binary counters together?

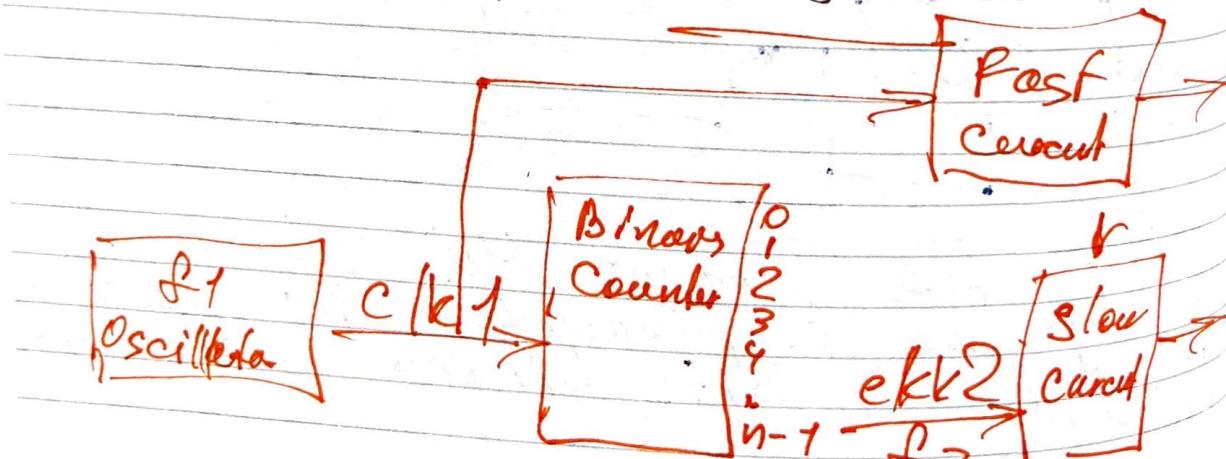
$$? \quad 16 \times 16 = 256$$

⑤. Clock frequency division  
 the process of generating a slow clock from a faster clock.

example, suppose we have two synchronous circuits driven by two separate clocks, CLK1 and CLK2. The synchronous circuit driven by CLK2 is slower than the one driven by CLK1, then we can use a clock divider circuit to generate the slow clock with the fast clock. Then both circuits can essentially be driven off the same clock, eliminating the need for two different oscillators.

Figure illustrates this concept. The frequency of the fast clock must be divided by a power of 2, then a simple binary counter can be used. If  $f_2 = f_1/2^n$ , then an n-bit binary counter will be sufficient.

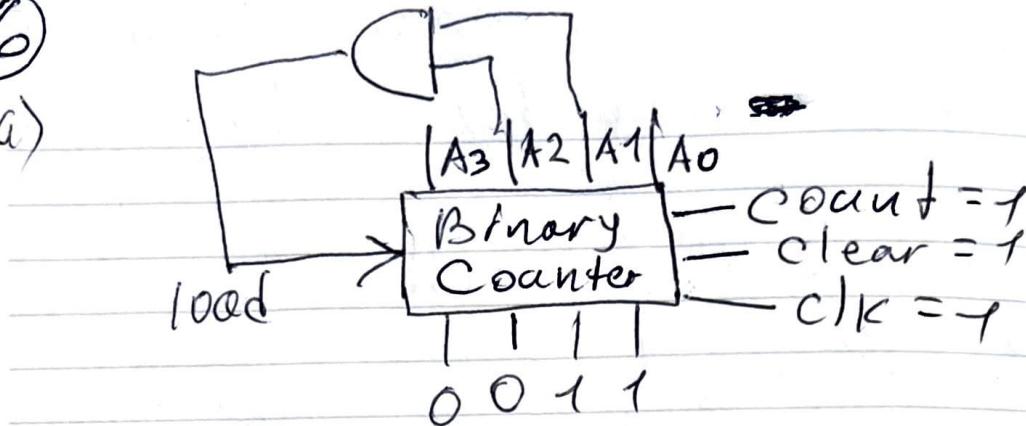
If you want to generate a 5.625 MHz clock from a 90 MHz clock, how many flip flops do you need?



$$f_2 = f_1/2^n \leftarrow \text{number of FF} \\ = 90/2^4 = 90/16 = 5.625$$

(6)

a)

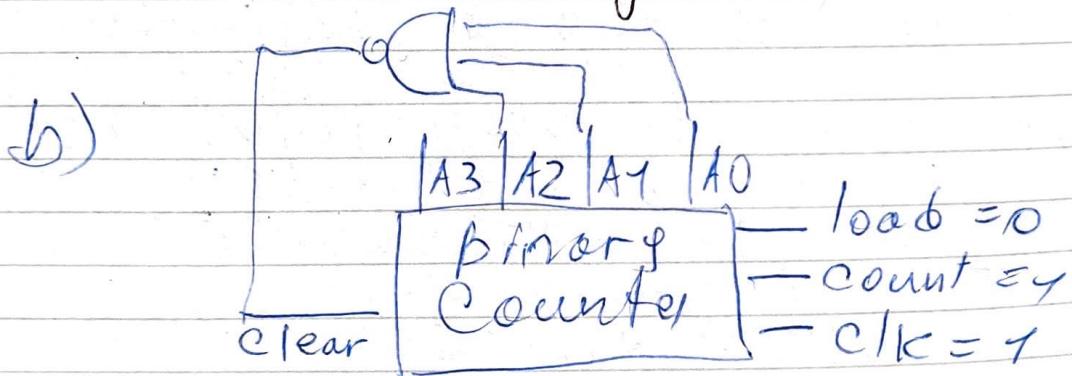


so if  $clk = 0$  is like electricity  
it won't work

if  $count = 0$ , then it won't count

\* if  $clear = 0$  probably won't work ch.

Once it gets to 6 it will load original data 0011 and if we'll increment overfill it gets to 0110.



using "clear" input

here it need to count to 0111 (7) not if always automatically load data from bottom

0000 to 0110

if it goes only till 6 it will clear but if it goes one more, then it will

⑦

a) $4K \times 16 = 2^2 \cdot 2^{10} = 2^{12}$	Address 12 lines	I/O data lines 16
b) $2G \times 8 = 2^1 \cdot 2^{30} = 2^{31}$	31	8
c) $16M \times 32 = 2^4 \cdot 2^{20} = 2^{24}$	24	32
d) $256K \times 64 = 2^8 \cdot 2^{10} = 2^{18}$	18	64

! Here we only work (decons tract) left part

! Power of that we get from previous one we do summation, not AND

The following memory units are specified by the number of words times the number of bits per word. How many addresser lines and input-output data lines are needed in each case.

⑧ Give the number of bytes stored in the memories listed above

$$1 \text{ byte} = 8 \text{ bits } 00000000$$

~~$34 \text{ bits} = 3 \text{ bytes and } 2 \text{ bits}$~~

! Based in definition in 4, second part is in bits, therefore we divided it by 8 bits (1 byte) in order to get in bytes

a)  $4K \times 16 = 4 \cdot K \times \frac{16}{8} = 4 \cdot K \times 2 \text{ bytes} = 8KB$

b)  $2G \times 8 = 2GB$

c)  $16M \times 32 = 64MB$

d)  $256K \times 64 = 2048KB = 2MB$

# ⑨ Memory Address

Binary	decimal	Memory
000000000000	0	1011011101
0000.....01	1	1001100011
00.....10	2	0001101000
1011011100	732	000001000111
111101111101	1021	1000101000
111111111100	1022	0000111000

word #732  
contains 1032

① 16 · K × 4 memory used concordant decoding split ~~→~~ decoder into X and Y-selection

a) size of each decoder

$$16 \cdot K = 2^4 \cdot 2^{10} = 2^{14}$$

based on power 14 we conclude that each decoder X and Y will have 7 inputs

And based on formula of n to  $2^n$

therefore decoder X size :  $7 \times 2^7$  aka  $7 \times 128$

and decoder Y size :  $7 \times 2^7$  aka  $7 \times 128$

b) Determine X and Y selection lines that are enabled when input is binary equivalent of 6000

$$\begin{array}{r} 6000 \\ \text{⑩} \end{array} = 0101110 | 1110000 \quad \begin{array}{l} \text{X} \\ \text{Y} \end{array} \quad \begin{array}{l} \text{addresses} \\ \#6000 \end{array}$$

adding extra  
for make  
7 bits

