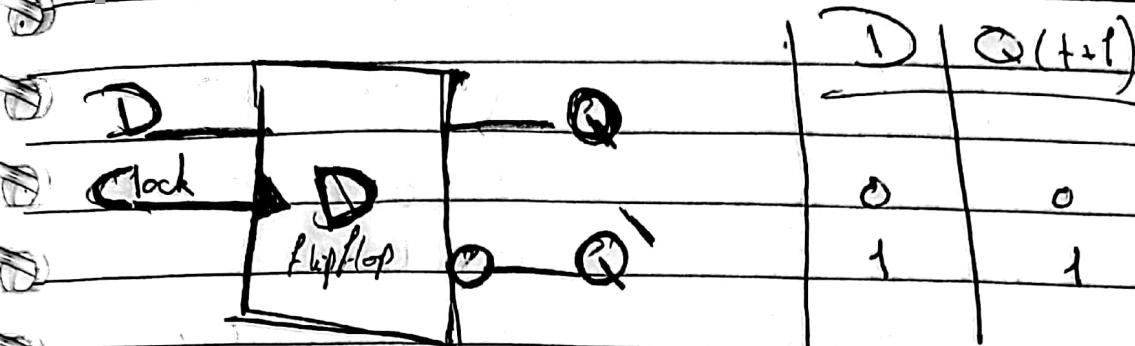
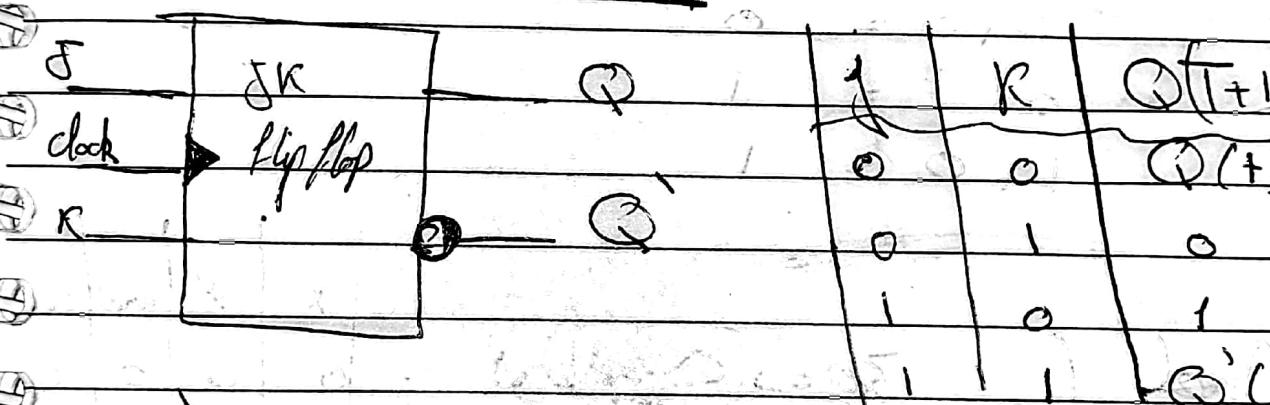


1 flip flop



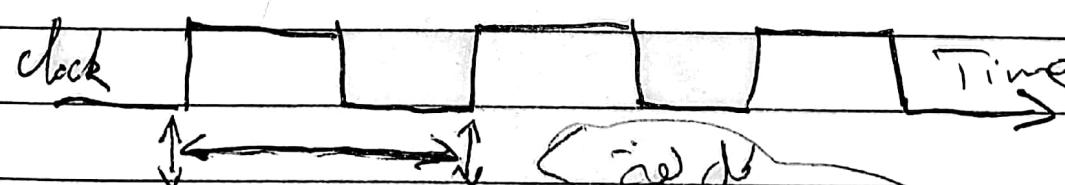
الشكل والتابعه في كل سره
out(Q) و Q' ← one edge, after clock

JR flip flop



Q' بار one edge zero ، KJ1 ، J1 لوار
one edge zero ، R one edge zero ، J1 لوار

D 1 (t+1)



new output
new input

clock 1st
cycle

Remember

PAGE _____

DATE _____

PgUp

$x \text{ XOR } y$

x

y

F

$x \text{ } y \text{ } F$

$0 \text{ } 0 \text{ } 0$

$0 \text{ } 1 \text{ } 1$

$1 \text{ } 0 \text{ } 1$

zero \leftarrow always

one \leftarrow never

Remember $1 + 1 = 10 \Rightarrow$ one N carry digit

$0 + 0 = 0$

~~1+0~~ $0 + 1 = 1$

~~1+0~~ $1 + 0 = 1$

And

zeros apply

zeros also give

$x \text{ } x$ and or

$0 \text{ } 0 \text{ } 0 \text{ } 0$

$0 \text{ } 0 \text{ } 1 \text{ } 0$

$0 \text{ } 1 \text{ } 0 \text{ } 1$

OR

one apply

one also give

Half Adder

Can 2 bits sum
class

$$\begin{array}{r} 1+1 \\ 0+0 \end{array}$$

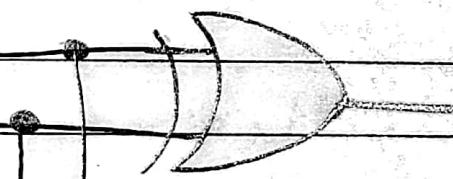
$$\begin{array}{r} 1+0 \\ 0+1 \end{array}$$

Sum 1 or 0, 2 Bits (Result)

X	Y	Sum	Carry	gates	Carry II
0	0	0	0		
0	1	1	0	new Sum	
1	0	1	0	Carry II	XOR
1	1	0	1	And	class

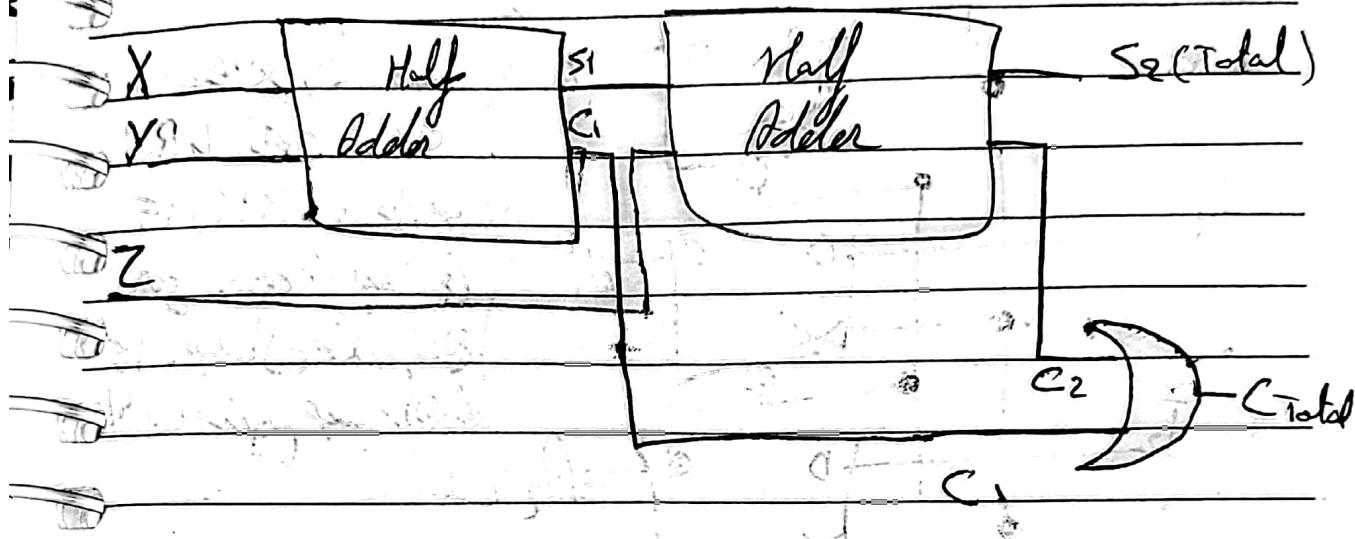
X

Y

Sum $\Rightarrow A \oplus B$ (XOR)Sum $\Rightarrow A \oplus B + C_{in}$ Carry $\Rightarrow A \cdot B + C_{in}$

Full Adder

2 Bits \rightarrow Sum & 2 Bits (Carry)



- جمل (Sum) و حمل (Carry) من كل Half Adder
- بناء على ذلك نجمع Sum من كل Half Adder
- نبني Carry من كل زان (Carry) و Sum (Total)
- OR من كل المدخلات
- بناء على ذلك نجمع Carry من كل زان (Carry) (Total)

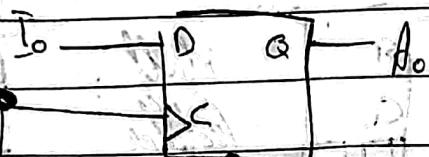
Registers

أمثلة علىRegisters (جهاز ذاكرة ويعمل على مدخلات و 输出) Flip-Flop 25

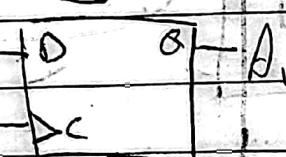
(4 Bit Register)

Bit من حيث

clock

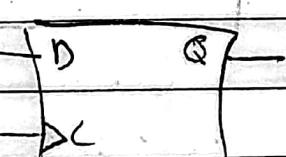


I₁



فمثلاً (عمر) 4Bit
وأقل (1111) 15

I₂



(0000) zero

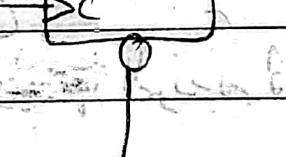
التي لها 15

I₃



التي لها 15

I₄



التي لها 15

clear

~~Registers~~ (Registers) 11 و 12

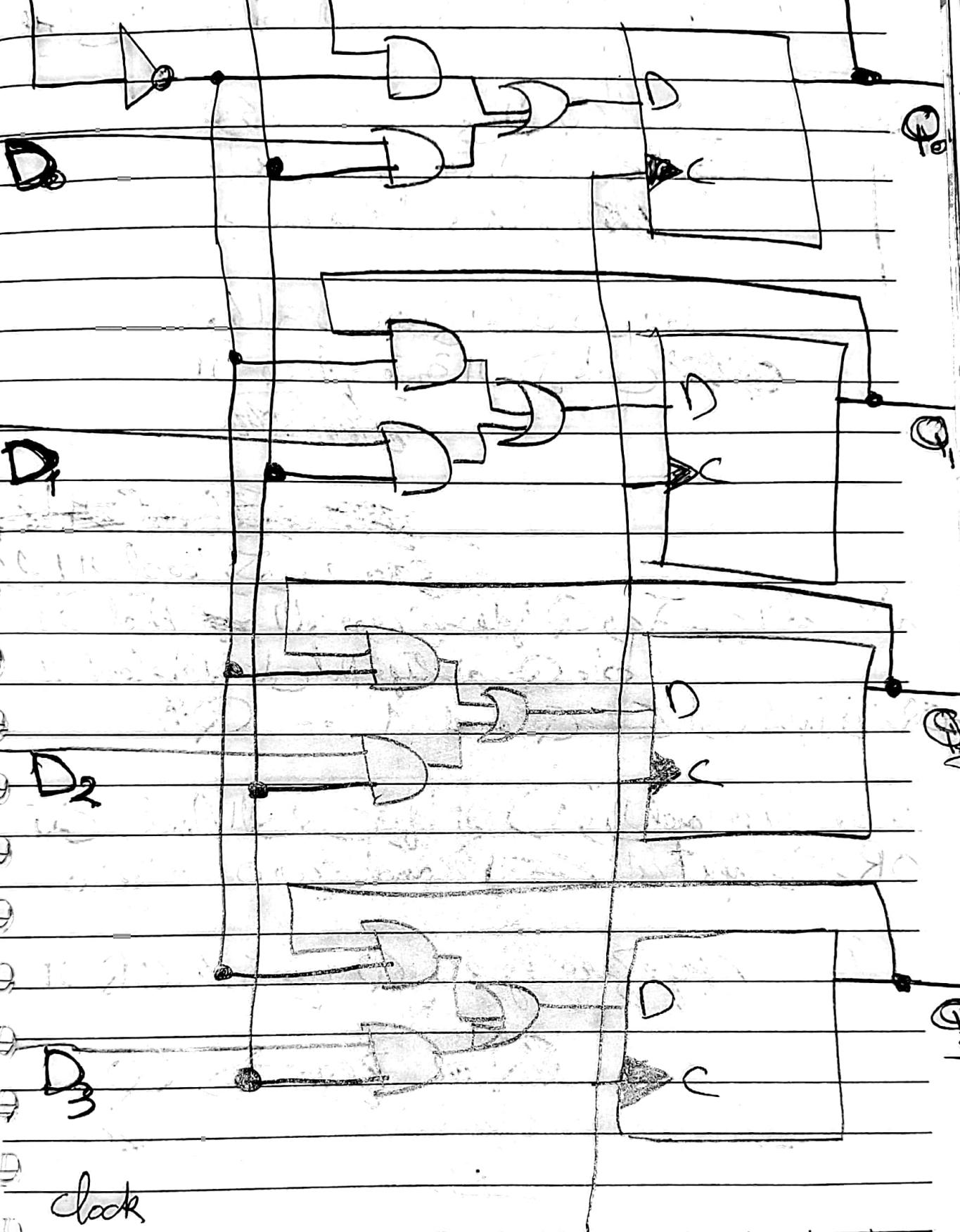
Registers (Registers) 11 و 12

Registers (Registers) 11 و 12

load Register

PAGE _____
DATE _____

load



Cont load Reg.

cycle

~~clock~~ العادي انه اول ما بال Reg. $A = I$

$(A = I)$ دخل input دخل output

Load

وكأنه بجز بدل الـ Q

~~clock~~ حتى لو الـ Q ما تغير

يكون فيه one

وكأنه في one في الحال

~~I~~ $I = 1$ في Q one

~~clock~~

~~load~~

$I = 0$ \rightarrow $Q = 0$ load

$I = 0$ \rightarrow $Q = 0$ \rightarrow And

$Q = 0$ \rightarrow Not \rightarrow Load

OR input اول $\equiv Q \text{ and } 1 \rightarrow Q$

الـ $I = 1$ \rightarrow $Q = 1$ And

OR input اول $\equiv D \text{ and zero} = 0$

عند zero $\rightarrow Q$ عند OR

$Q \text{ or zero} \Rightarrow Q$

1 Cont (all) Rog -

ion, general (II)

اولاً وتعتبر الـ **final** library **final** **load** **load** **final** **final** **final** **final** **final** **final** **final**

OR if input $\omega_i \in Q$ and $z_{\omega_i} = z_{\text{zero}}$

enjoy ~~the~~ load ~~on~~ II, DJI like, will load II
- DJI like

OR) if input is $i \in D$ and one = $\neg D$

llegó a su casa D llegó a su casa OR JI

~~Dar zero = 1~~

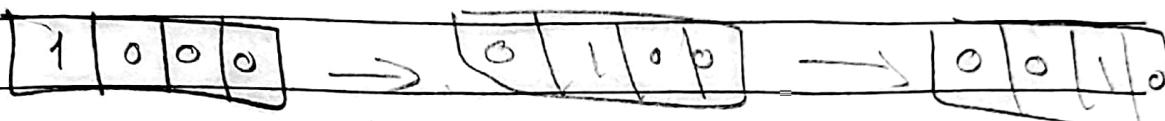
~~Interfacing of OR, 2 And
2x1 Multiplexer~~ using [jst] -

4-Bit Shift Register

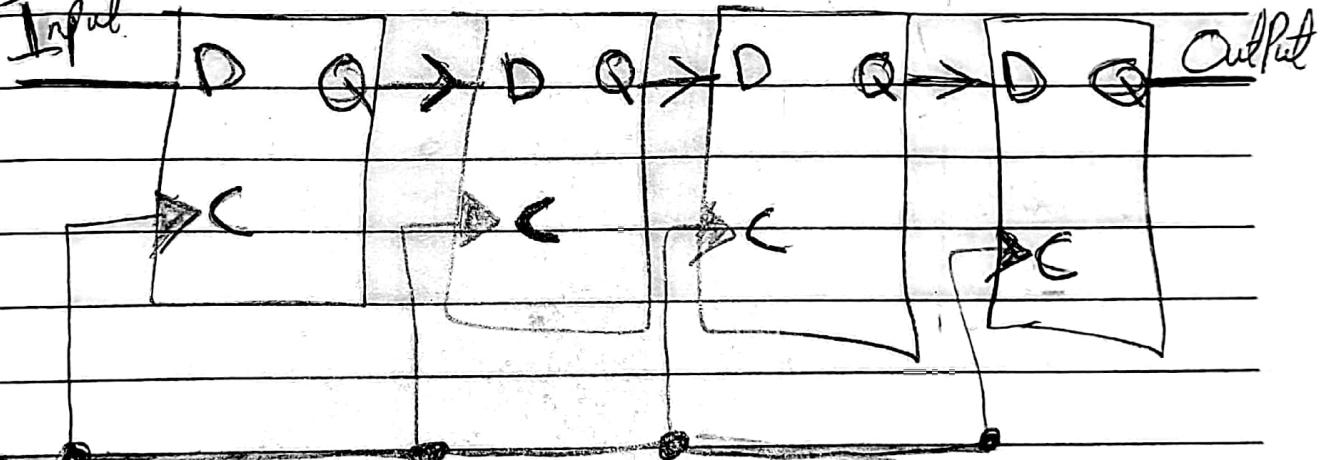
PAGE
DATE

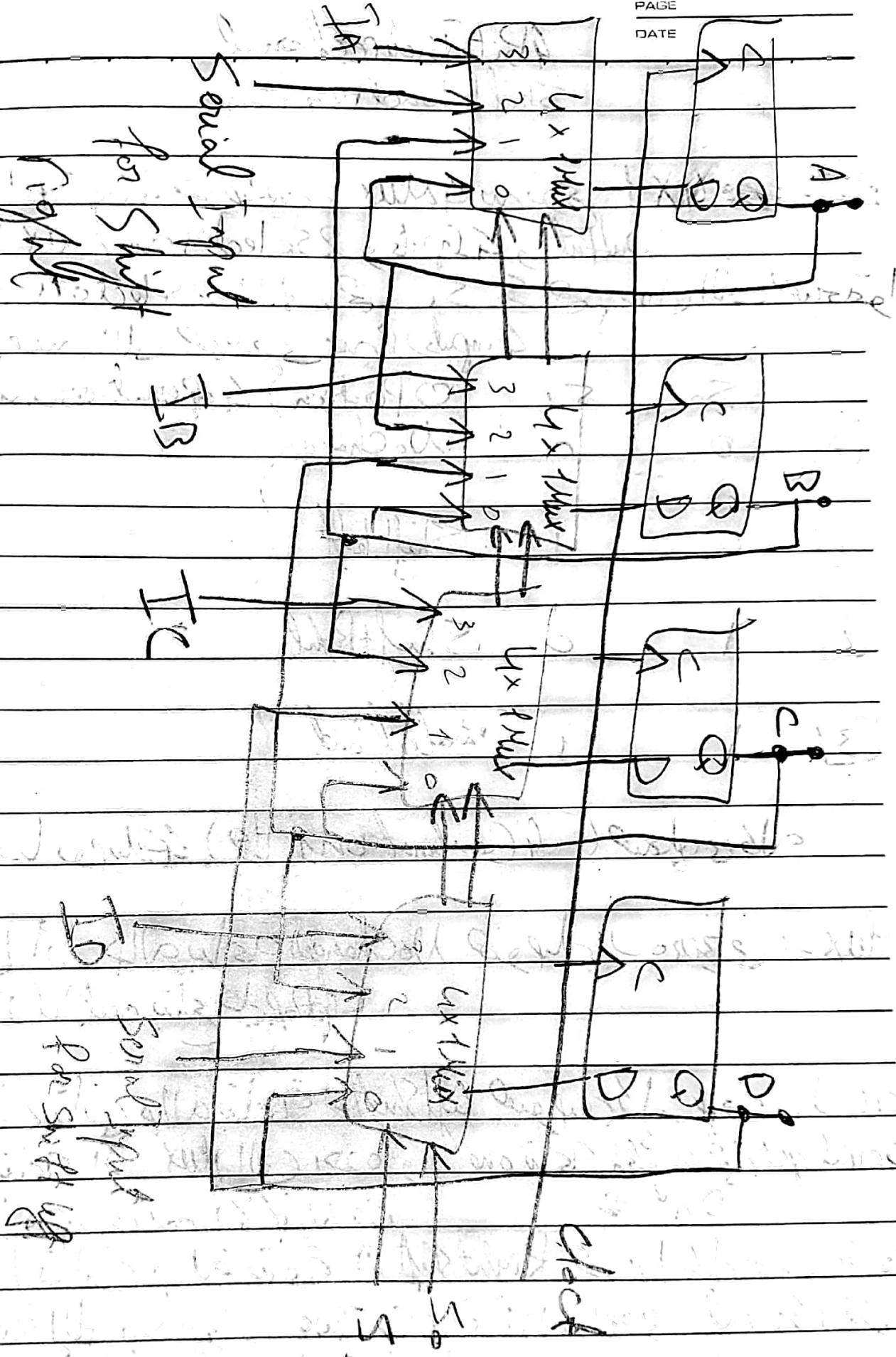
في كل خطوة، يدخل دخول D، ويرسل Q إلى خطوة التالية، D، وهذا ينطبق على كل خطوة، حتى النهاية، حيث يخرج الناتج من الخطوة الأخيرة.

1



Input





4 Bit Bidirectional Shift Load Register

PAGE
DATE

~~4X1~~ using 4 MUX - using 4 flipflops line
1 Output, 4 Inputs, 2 Selectors in MUX
1 S₀, S₁ 2 S₀, S₁ 2 Selectors
4 inputs from 4 flipflops
S₀ S₁ Operation 4 operation logic
0 0 No Change

[1] 0 1 Shift left

[2] 1 0 Shift Right

[3] 1 1 Parallel Load

all (4 combinations) 4 flipflops

MUX (zero) No change (parallel output)
flipflop output

one (left shift) serial input (D₁)
Serial input (D₁) one (0) MUX (parallel output)

two (right shift) Serial input (D₁) two (1) MUX (parallel output)

الخطوة ٤: Parallel load (الخطوة الأولى في الـ ٤)

8 Input might be $M_{11}X_{11}$ (three J_5)

لارج فونت میں اسکے لئے $\text{load } \text{arzi}$ کا ڈیکل فنی F.F فیکٹری میں سے ایک سیکھیں۔

Right Shift Operator
↳ Bitwise

left Shift \leftarrow time
right Shift \rightarrow space