

lecture 5

Sequential circuits 1

- RS latch

Assignment Project Exam Help

- D latch

<https://eduassistpro.github.io/>

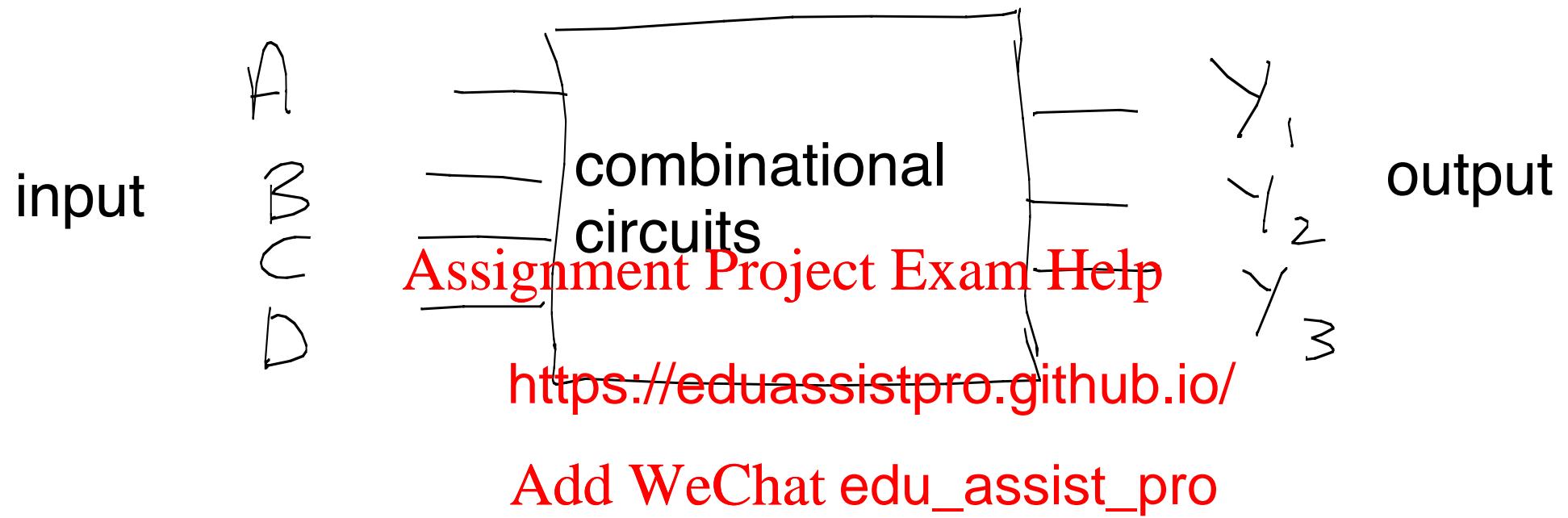
- flipflops (D)

Add WeChat edu_assist_pro

- registers

January 25, 2016

last week....



Add WeChat edu_assist_pro

- truth tables and circuit diagrams
- 0 and 1 signals are (voltage) values on wires
- circuits take time to "compute" e.g. carries in addition

this week....

clock C

input

sequential

Assignment Project Exam Help output

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

⌚ combinatorial circuits + *memory*

⌚ synchronized by a clock C

Memory (two kinds)

- write it down
[Assignment](#) [Project](#) [Exam](#) [Help](#)
<https://eduassistpro.github.io/>
- repeat it to yourself (fee)
[Add WeChat](#) [edu_assist_pro](#)

Sequential circuits use the latter.

Latch

Assignment Project Exam Help

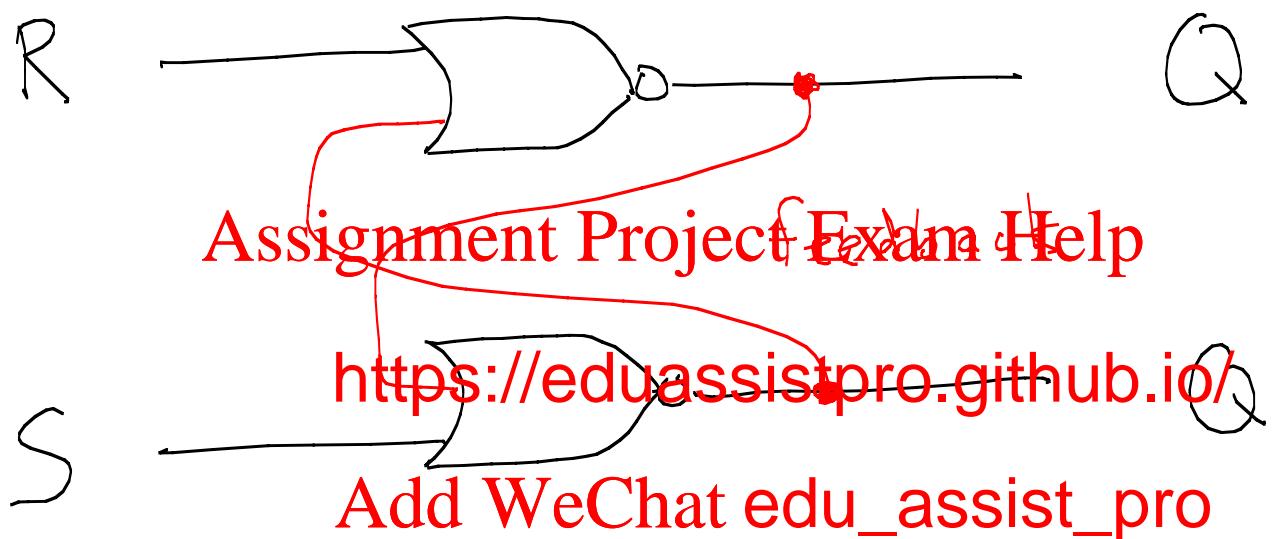
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

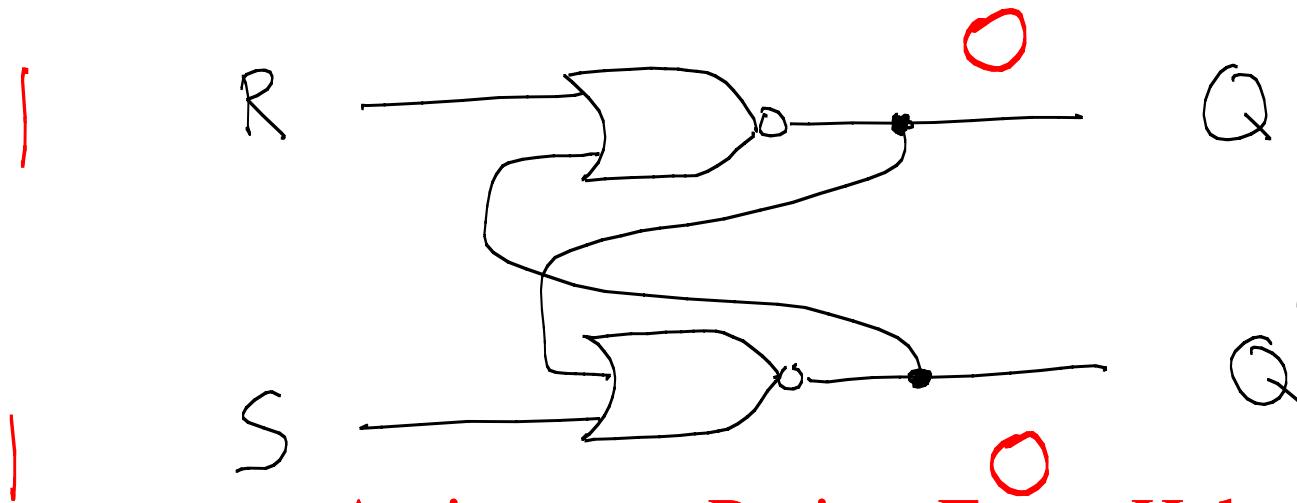
definition (wikipedia); "... a type of ... fastener that is used to join two objects or surfaces together while allowing for the regular ... separation of the surfaces"

Latches are often (but not always) used to block paths, e.g. close doors.

RS latch ('reset' 0, 'set' 1)



A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

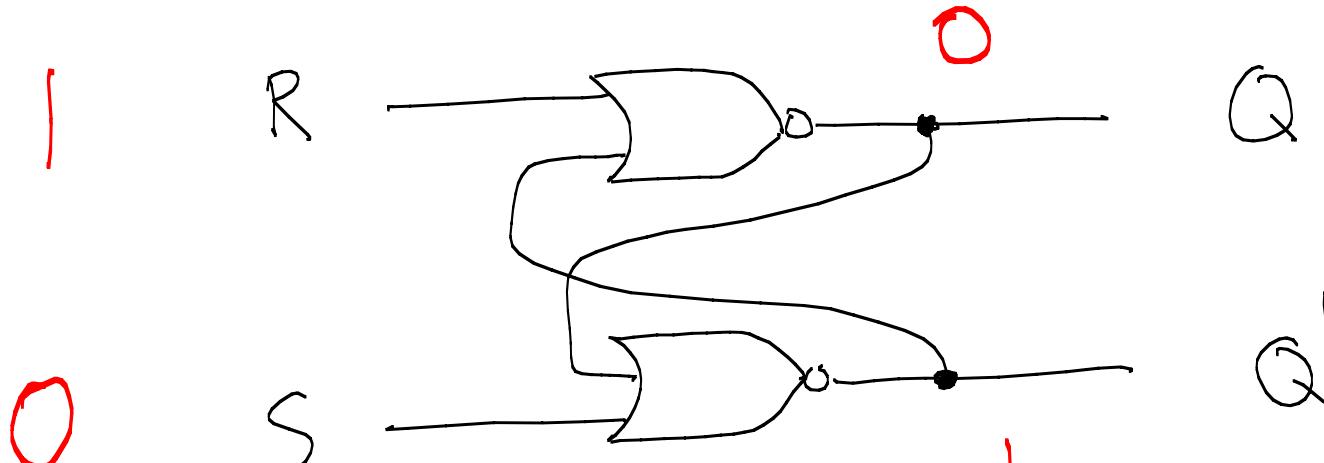


Assignment Project Exam Help

$R = S = 1$
inputs will not
be allowed.

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

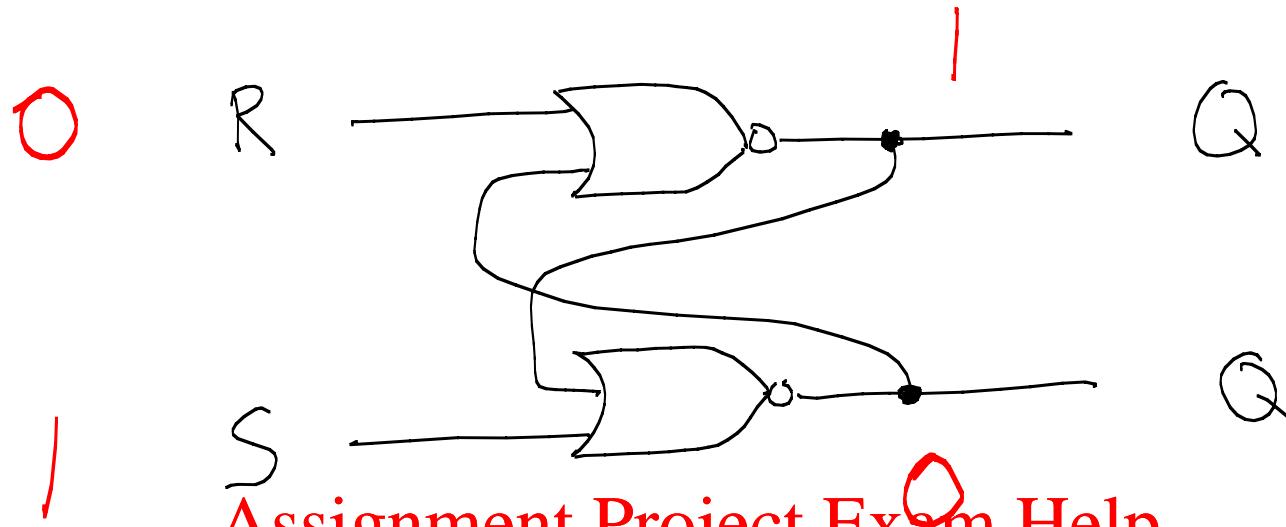


Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Q

R

Q

I

Q

$I \rightarrow Q$

S

I

Q'

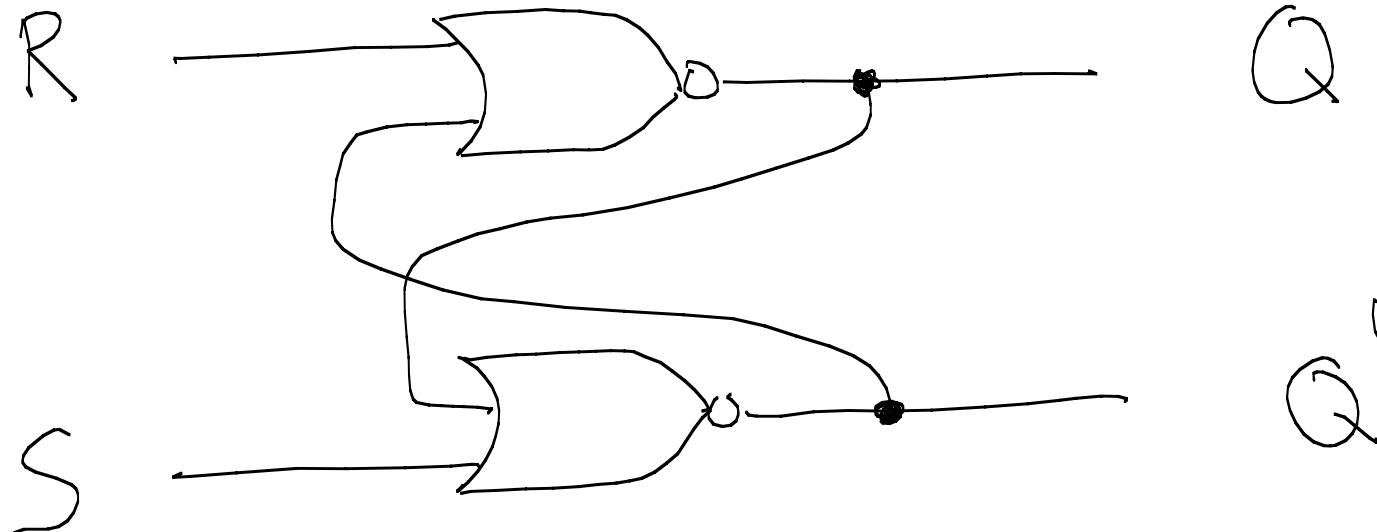
Assignment Project Exam Help

Q, \bar{Q}

<https://eduassistpro.github.io/>

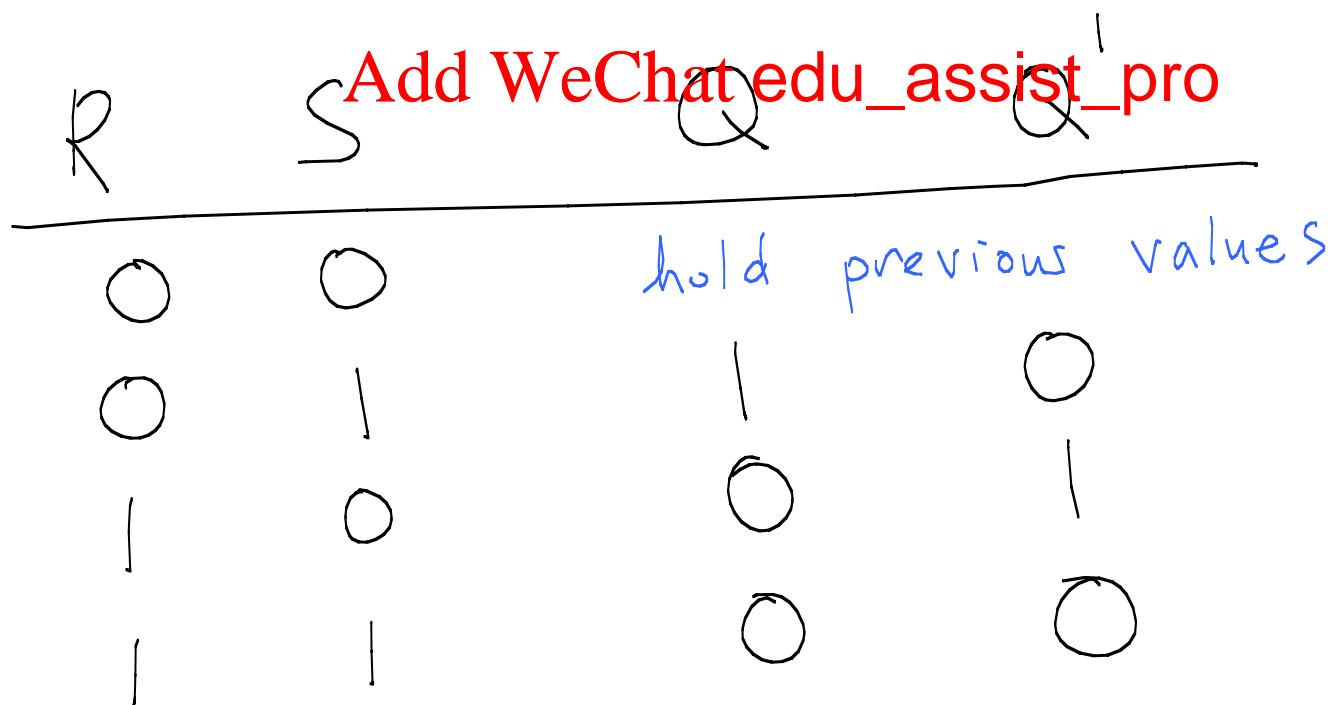
values do not change
Add WeChat edu_assist_pro

A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

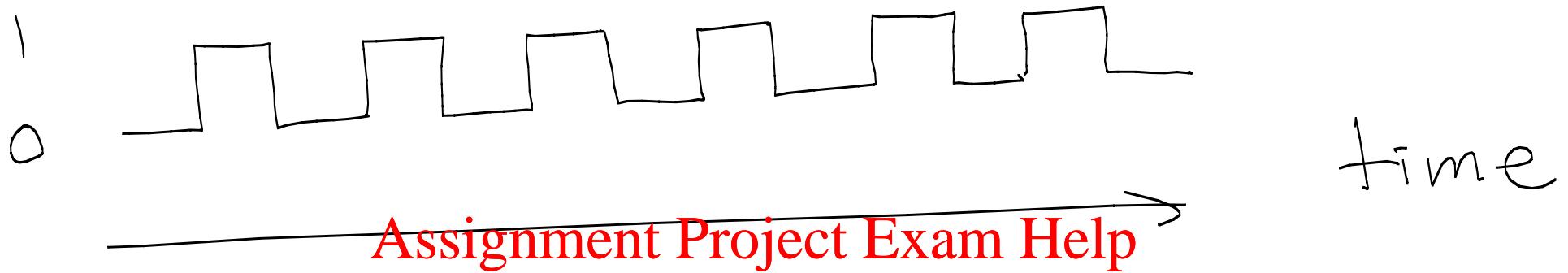


Assignment Project Exam Help

<https://eduassistpro.github.io/>



Clock



Assignment Project Exam Help

<https://eduassistpro.github.io/>

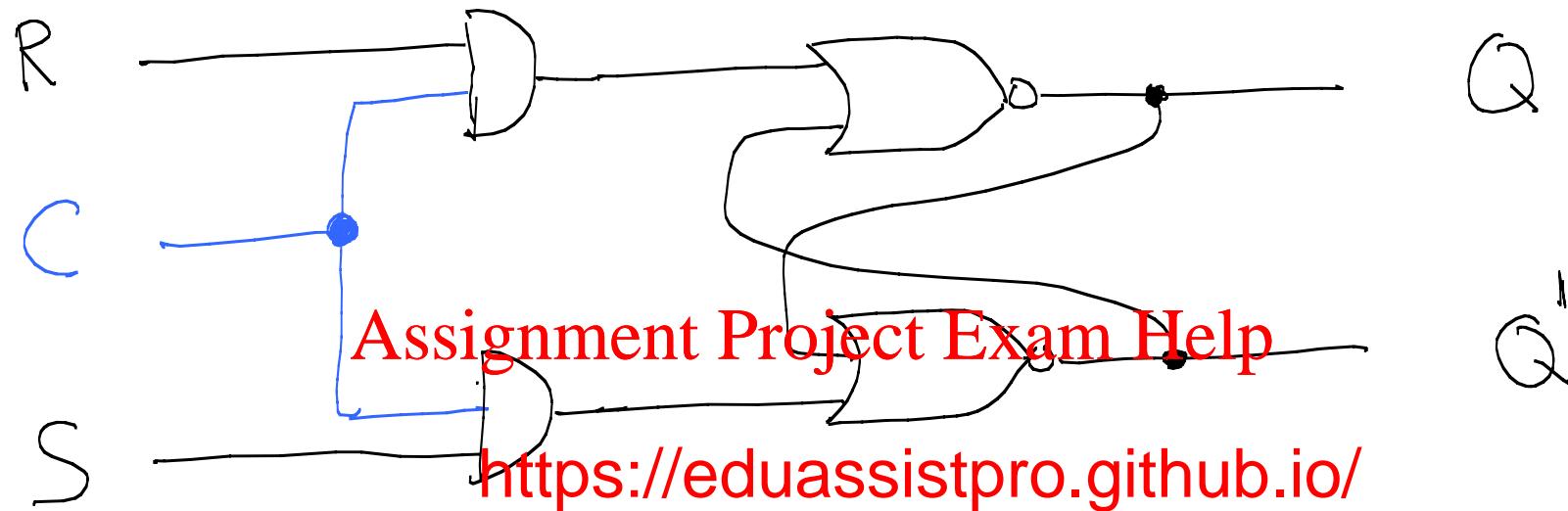
Add WeChat edu_assist_pro

- electronic implementation uses "crystal oscillator"

https://en.wikipedia.org/wiki/Clock_signal

- typical clock speed is in gigahertz (10^9 cycles/sec)

Clocked RS latch



Add WeChat edu_assist_pro

$$C = 0$$

R	S	Q	Q'
0	0	0	1
0	1	1	0
1	0	0	1
1	1	0	1

hold

$$C = 1$$

R	S	Q	Q'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

hold

Example

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

$C = 0$

R	S	Q	Q'
0	0		
0	1		
1	0		
1	1		

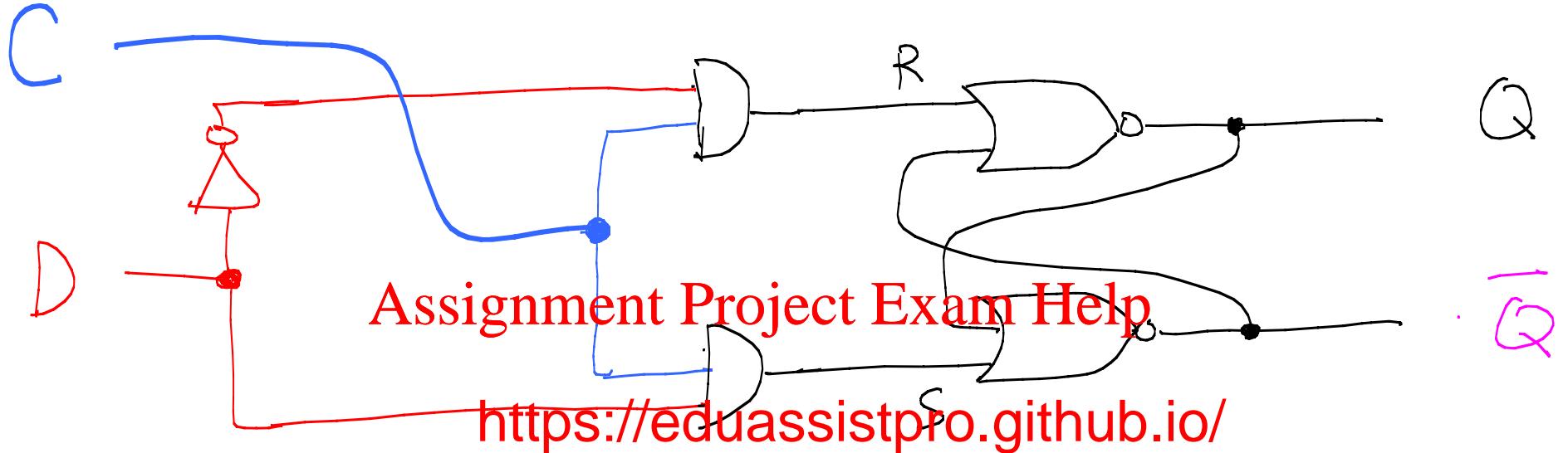
hold

$C = 1$

R	S	Q	Q'
0	0		
0	1		
1	0		
1	1		

hold

D latch ("D" is for data)

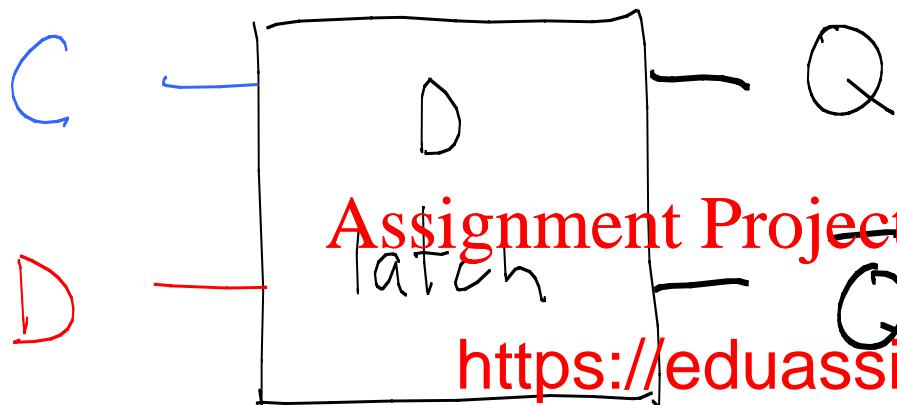


Add WeChat edu_assist_pro
What does this circuit do ?

when $C = 1$, $D = 1 \Rightarrow Q = 1, \bar{Q} = 0$
 $D = 0 \Rightarrow Q = 0, \bar{Q} = 1$

When $C = 0$, hold values of Q, \bar{Q}

D latch



Assignment Project Exam Help

<https://eduassistpro.github.io/>



Add WeChat edu_assist_pro

C = 0 holds values in D latches. (Read only)

C = 1 allows values in D latches to go through.
(Read and write).

Example:

Suppose we used D latches to store 8 bit numbers A and B.

Suppose we added A and B using the circuit below

and wrote the new value back into B. Would th

work? No, because <https://eduassistpro.github.io/>

when C = 1 there woul

be no control over timing [Add WeChat edu_assist_pro](#)

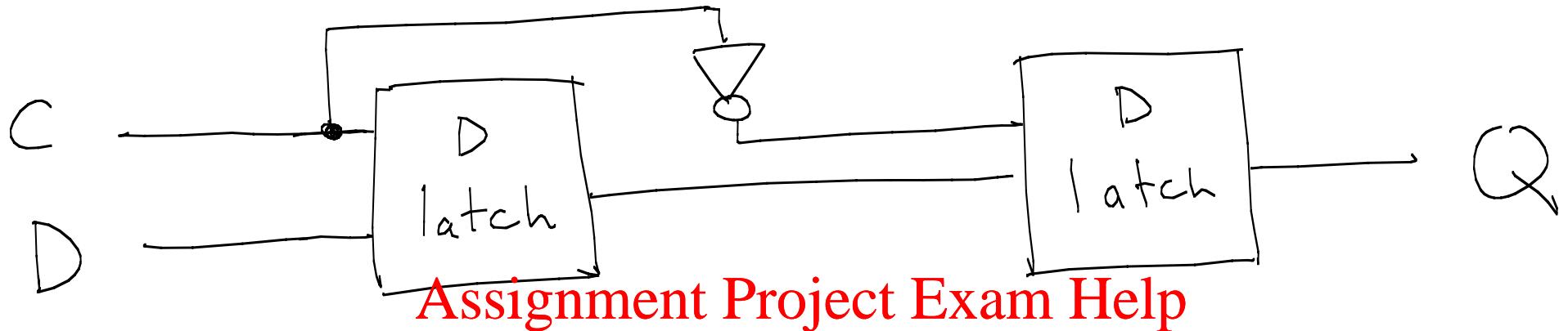
and we could loop

through multiple times

within a single clock pulse

(while C = 1).

D flip flop



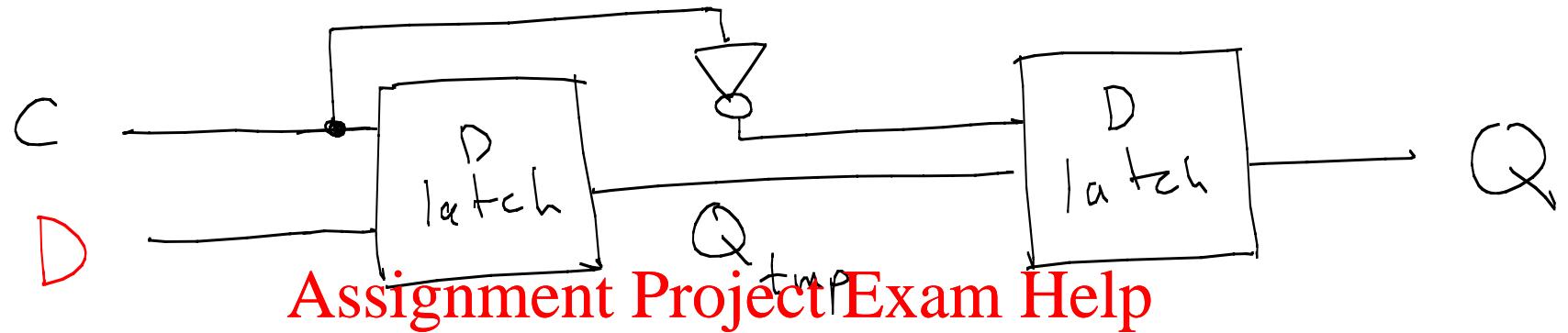
$C = 1$

Write D v <https://eduassistpro.github.io/>
Q doesn't change
Add WeChat edu_assist_pro

$C = 0$

Stop writing D into first D latch.
The D value from first D latch is written into second, so Q gets a possibly new value.

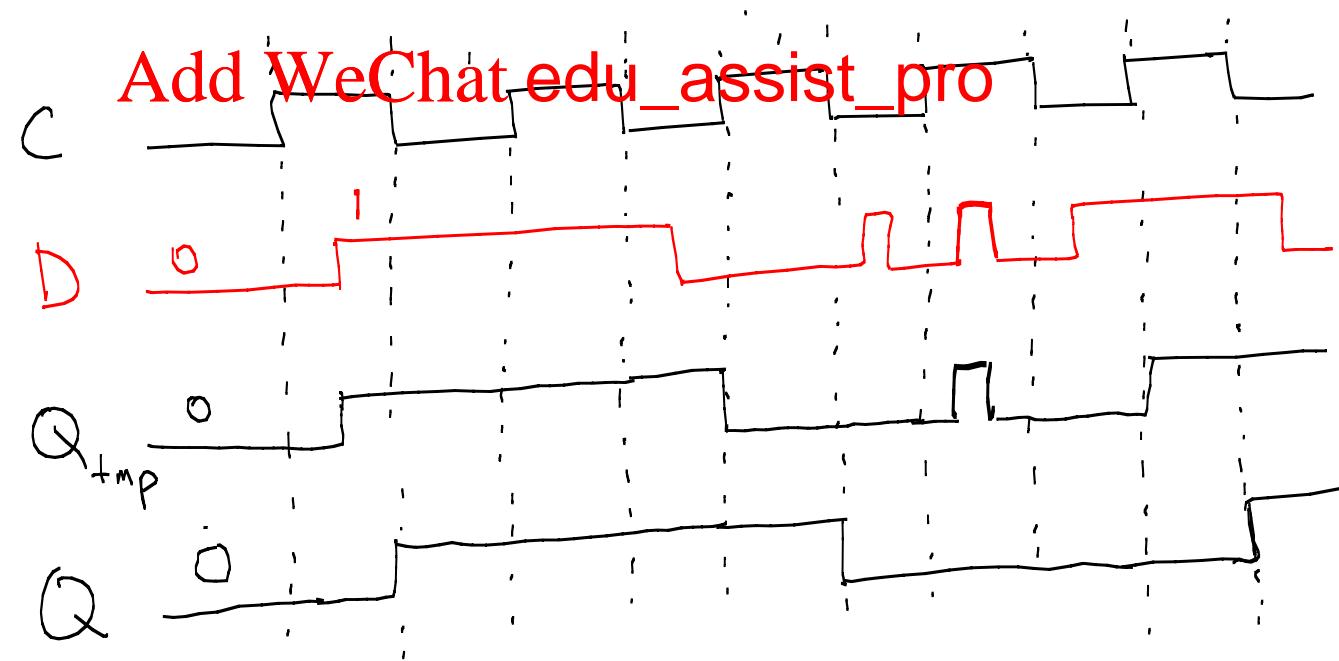
D flip flop ("falling edge triggered")



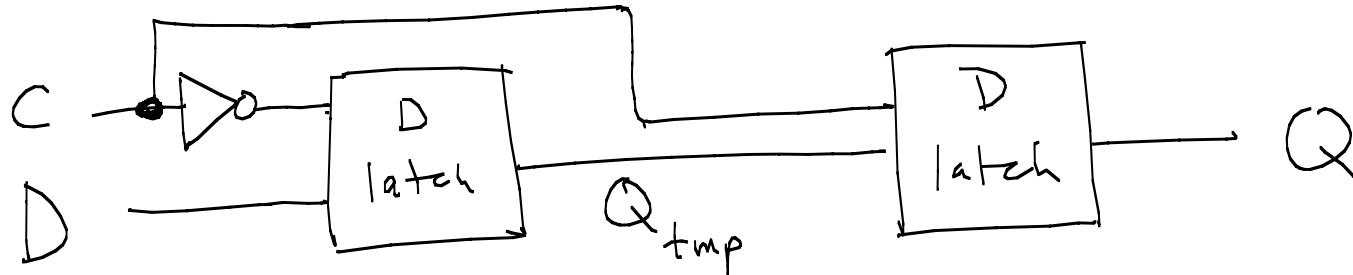
<https://eduassistpro.github.io/>

Add WeChat **edu_assist_pro**

example



D flip flop ("rising edge triggered")



By putting the inversion on the first D latch, we would make Q change its value on the falling edge. There is no advantage to this, so I will use this next

Add WeChat edu_assist_pro

Clock cycle must be long enough to allow all gates to stabilize.

Clock synchronizes all flipflops, allowing us to treat time as a sequence of discrete read/write steps (hence 'sequential circuit')

From now on,

Assignment Project Exam Help

- we ignore all variations
e.g. carries in the adder <https://eduassistpro.github.io/>
- we work only with D flipflops (no J-K)

Add WeChat edu_assist_pro

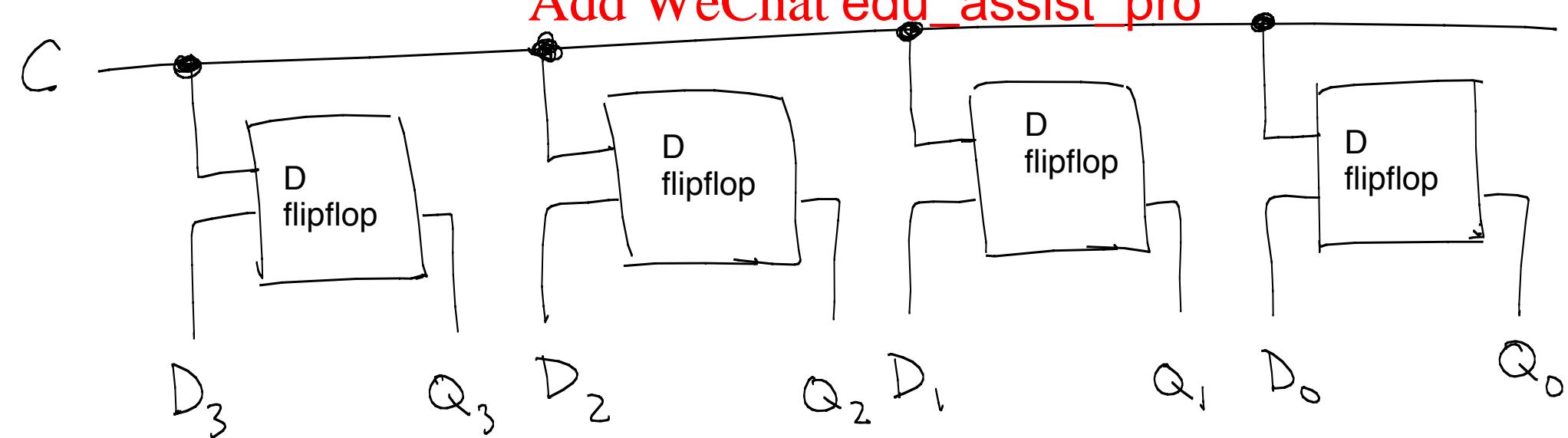
Register

(set of flipflops that are read/written together)

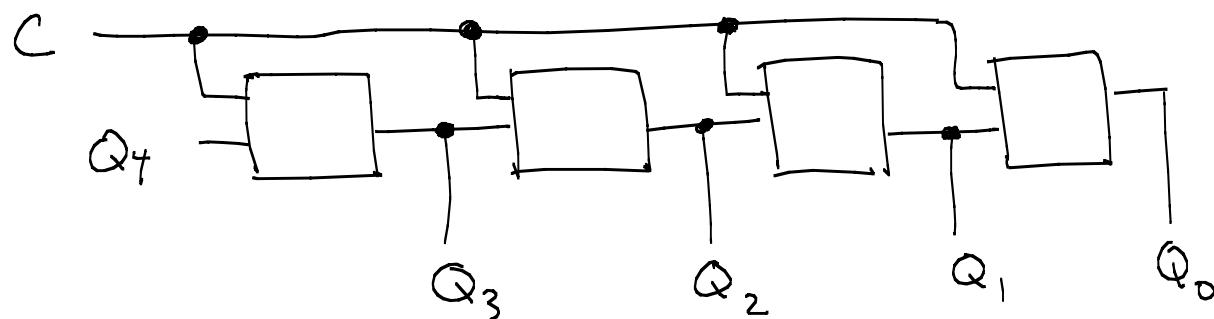
Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



Shift Right Register (falling edge)



Assignment Project Exam Help

Example:

suppose at $t = 0$.

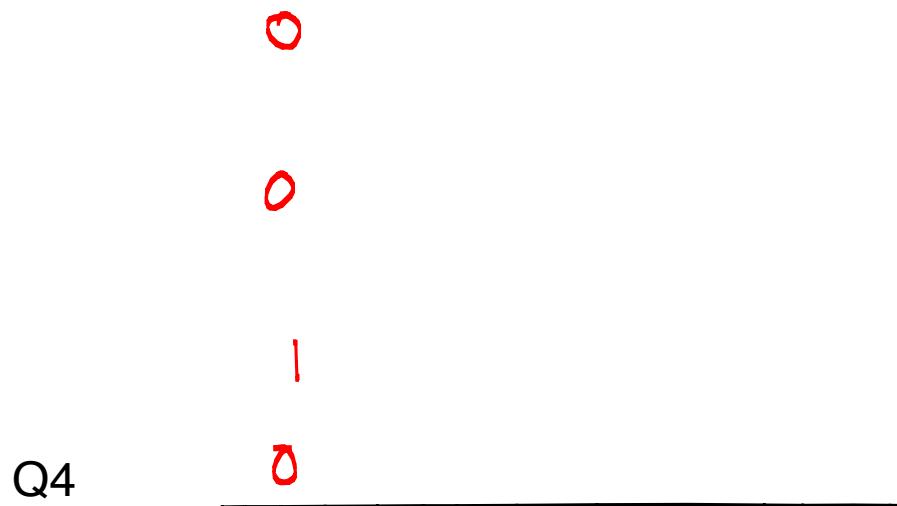
$(Q_4, Q_3, Q_2, Q_1, Q_0)$
is $(0, 1, 0, 0, 1)$

Q_4 remains at 0 for the
five clock pulses shown.

What happens at each
falling edge of clock ?

C <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



Shift Right Register

We can make Q4 have other values e.g. D (variable), 1, 0, Q0.

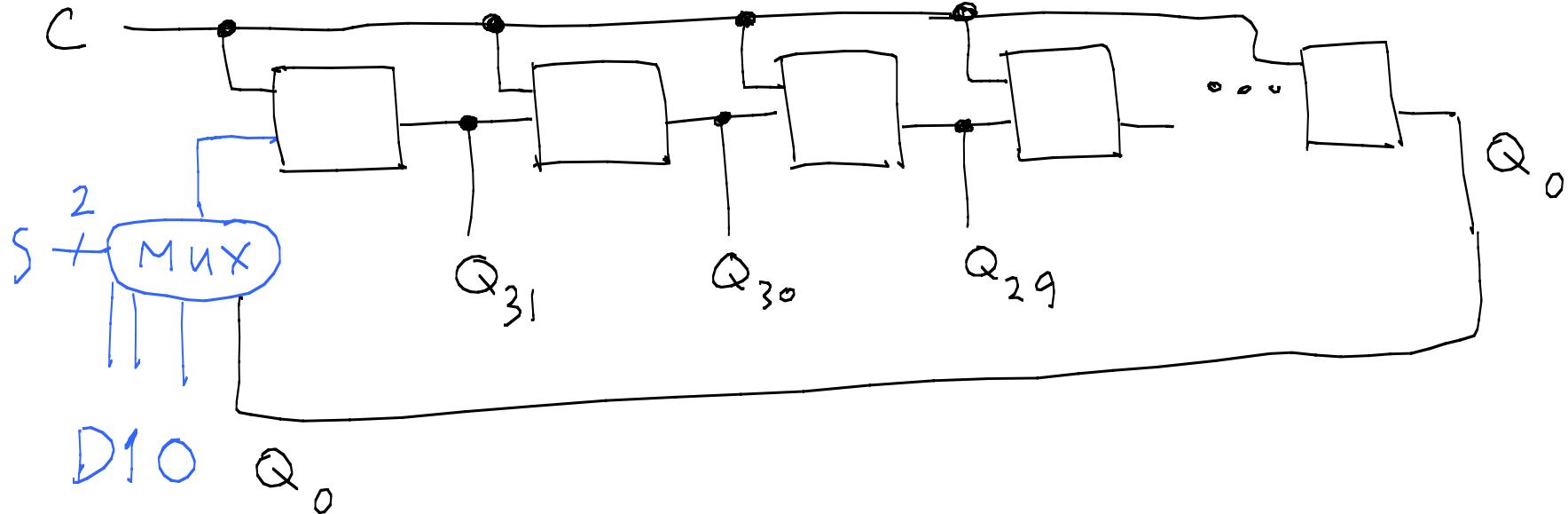
We can then *select* which of these gets put into the MSB.

Assignment Project Exam Help

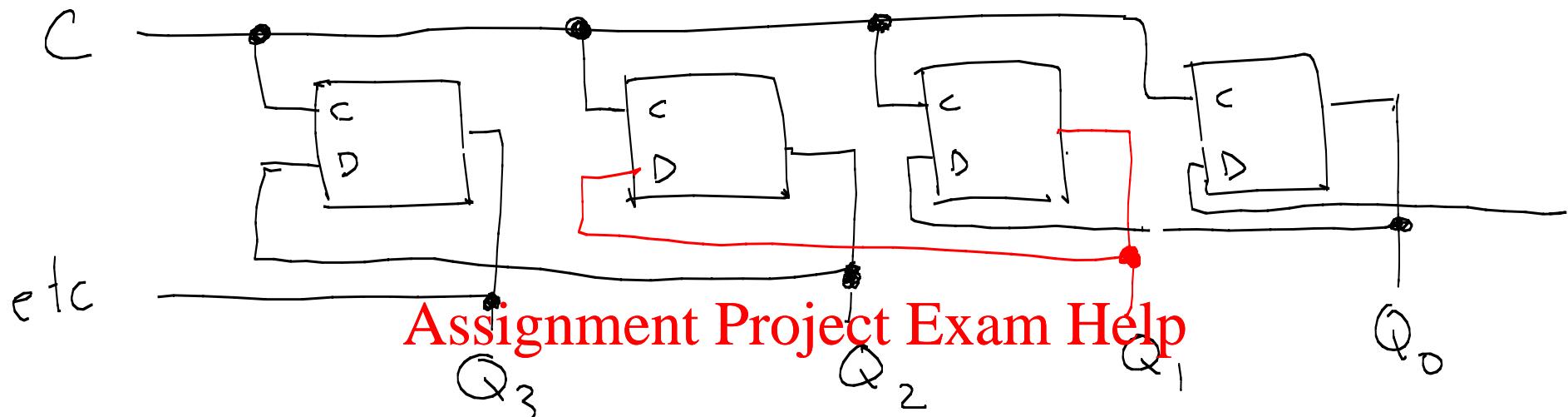
In this example the multiplexor selects what gets input at the left side, so the clock it is then gets prop

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



Shift Left Register

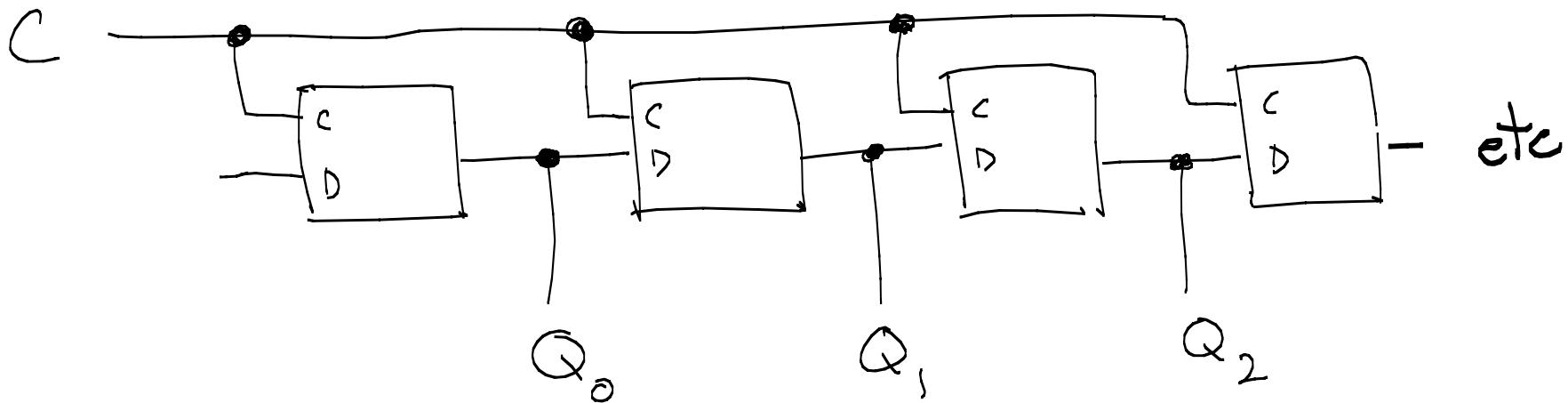


Assignment Project Exam Help

<https://eduassistpro.github.io/>

Alternatively, physically order the flipflops in
positive order

Add WeChat edu_assist_pro



Select from:

- shift left
- shift right
- write data
- clear

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat-edu-assist_pro

