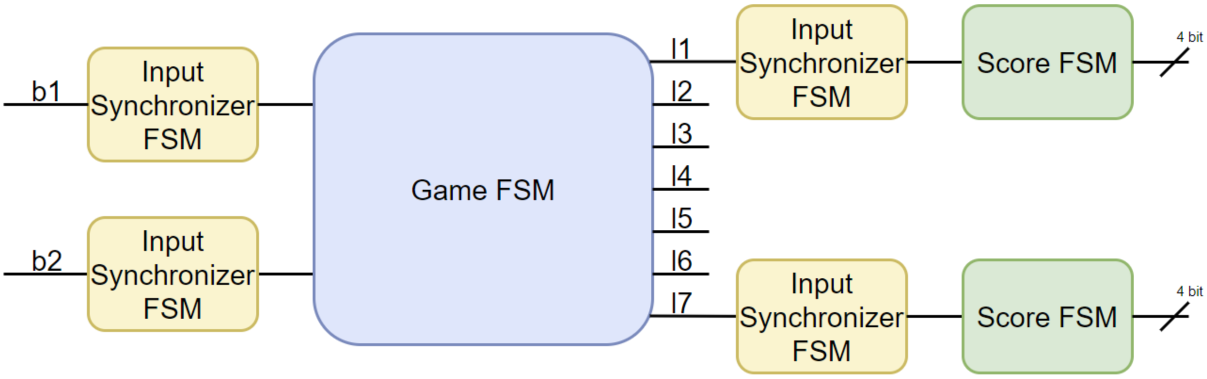
CSE 232 SPRING 2020

PROJECT 1

ABDULLAH CELIK

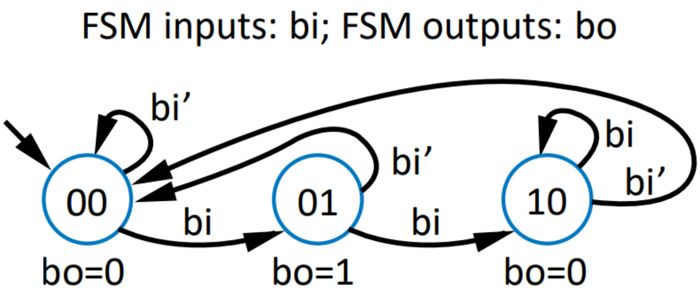
171044002

SYSTEM



Input Synchronizer FSM

State Diagram



Truth Table

Inputs			Outputs		
s1	**s0**	**bi**	**n1**	**n0**	**bo**
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	0	0

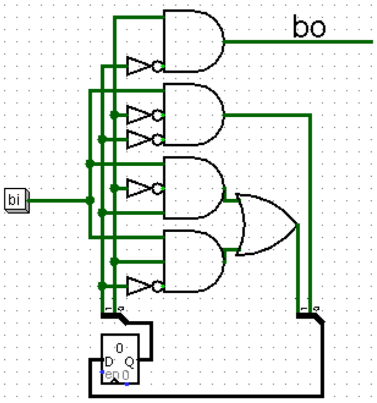
Boolean Expressions

n1 = s1’s0bi + s1s0’bi

n0 = s1’s0’bi

bo = s1’s0bi’ + s1’s0bi = s1’s0

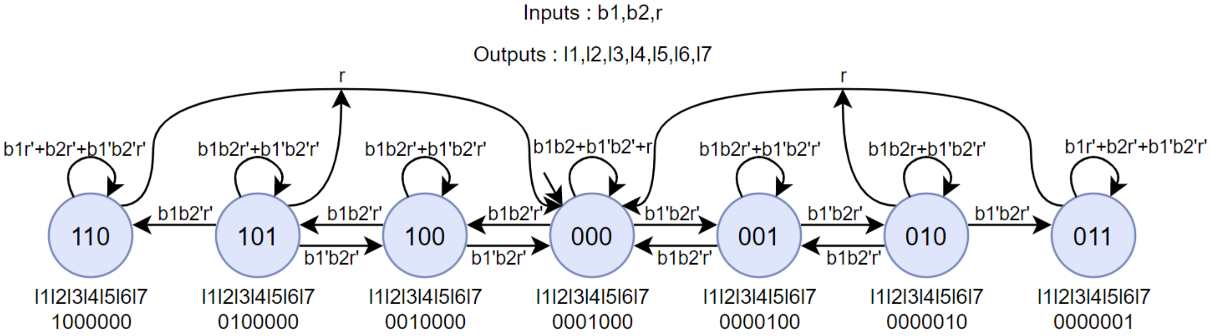
Circuit on Logisim



Game FSM

State Diagram

Note : Input b1 and b2 always output 1 clock timelength 1 output. Input synchronizer provides this.



Truth Table

Game Combinational Logic															
Inputs						Outputs									
s2	**s1**	**s0**	**b1**	**b2**	**r**	**n2**	**n1**	**n0**	**l1**	**l2**	**l3**	**l4**	**l5**	**l6**	**l7**
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	X	X	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	1	0	0	0	1	0	0	0	0	1	0	0
0	0	1	X	X	1	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0
0	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0
0	1	0	X	X	1	0	0	0	0	0	0	0	0	1	0
0	1	1	X	X	0	0	1	1	0	0	0	0	0	0	1

0	1	1	X	X	1	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0
1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0
1	0	0	X	X	1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0
1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0
1	0	1	1	1	0	1	0	1	0	1	0	0	0	0	0
1	0	1	X	X	1	0	0	0	0	1	0	0	0	0	0
1	1	0	X	X	0	1	1	0	1	0	0	0	0	0	0
1	1	0	X	X	1	0	0	0	1	0	0	0	0	0	0
1	1	1	X	X	X	0	0	0	0	0	0	0	0	0	0

Boolean Expressions

- n2

When n2 output is 1					
Inputs					
s2	**s1**	**s0**	**b1**	**b2**	**r**
0	0	0	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	X	X	0

After some logical algebra

When n2 output is 1					
Inputs					
s2	**s1**	**s0**	**b1**	**b2**	**r**
0	0	0	1	0	0
1	0	0	X	0	0
1	0	0	1	1	0
1	0	1	X	X	0
1	1	0	X	X	0

n2 = s2’s1’s0’b1b2’r’ + s2s1’s0’b2’r’ + s2s1’s0’b1b2r + s2s1’s0r’ + s2s1s0’r’
n2 = s1’s0’b2’r’(s2’b1 + s2) + s2s1’s0’b1b2r + s2s1’s0r’ + s2s1s0’r’
n2 = s2s1’s0’b2’r’ + s1’s0’b1b2’r’ + s2s1’s0’b1b2r + s2s1’s0r’ + s2s1s0’r’
n2 = s2s1’s0’r’(b2’ + b1b2) + s1’s0’b1b2’r’ + s2s1’s0r’ + s2s1s0’r’
n2 = s2s1’b1s0’r’ + s2s1’s0’b2’r’ + s1’s0’b1b2’r’ + s2s1’s0r’ + s2s1s0’r’
n2 = s2s0’r’(s1’b1 + s1) + s2s1’r’(s0’b2’ + s0) + s1’s0’b1b2’r’
n2 = s2s1s0’r’ + s2s0’b1r’ + s2s1’s0r’ + s2s1’b2’r’ + s1’s0’b1b2’r’

- n1

When n1 output is 1					
Inputs					
s2	**s1**	**s0**	**b1**	**b2**	**r**
0	0	1	0	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	X	X	0
1	0	1	1	0	0
1	1	0	X	X	0

After some logical algebra

When n1 output is 1					
Inputs					
s2	**s1**	**s0**	**b1**	**b2**	**r**
0	0	1	0	1	0
0	1	0	0	0	0
0	1	0	X	1	0
0	1	1	X	X	0
1	0	1	1	0	0
1	1	0	X	X	0

n1 = s2’s1’s0b1’b2r’ + s2’s1s0’b1’b2’r’ + s2’s1s0’b2r’ + s2’s1s0r’ + s2s1’s0b1b2’r’ + s2s1s0’r’

n1 = s2’s1’s0b1’b2r’ + s2’s1s0’r’(b1’b2’ + b2) + s2’s1s0r’ + s2s1’s0b1b2’r’ + s2s1s0’r’

n1 = s2’s1’s0b1’b2r’ + s2’s1s0’b1’r’ + s2’s1s0’b2r’ + s2’s1s0r’ + s2s1’s0b1b2’r’ + s2s1s0’r’

n1 = s2’s0r’(s1’b1’b2 + s1) + s1s0’r’(s2’b1’ + s2) + s2’s1s0’b2r’ + s2s1’s0b1b2’r’

n1 = s2’s0b1’b2r’ + s2’s1s0r’ + s2s1s0’r’ + s1s0’b1’r’ + s2’s1s0’b2r’ + s2s1’s0b1b2’r’

n1 = s2’s0b1’b2r’ + s2’s1r’(s0 + s0’b2) + s2s1s0’r’ + s1s0’b1’r’ + s2s1’s0b1b2’r’

n1 = s2’s0b1’b2r’ + s2’s1s0r’ + s2’s1b2r’ + s2s1s0’r’ + s1s0’b1’r’ + s2s1’s0b1b2’r’

- n0

When n0 output is 1					
Inputs					
s2	**s1**	**s0**	**b1**	**b2**	**r**
0	0	0	0	1	0
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	X	X	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0

After some logical algebra

When n0 output is 1					
Inputs					
s2	**s1**	**s0**	**b1**	**b2**	**r**
0	X	0	0	1	0
X	0	1	0	0	0
X	0	1	1	1	0
0	1	0	1	0	0
0	1	1	X	X	0
1	0	0	1	0	0

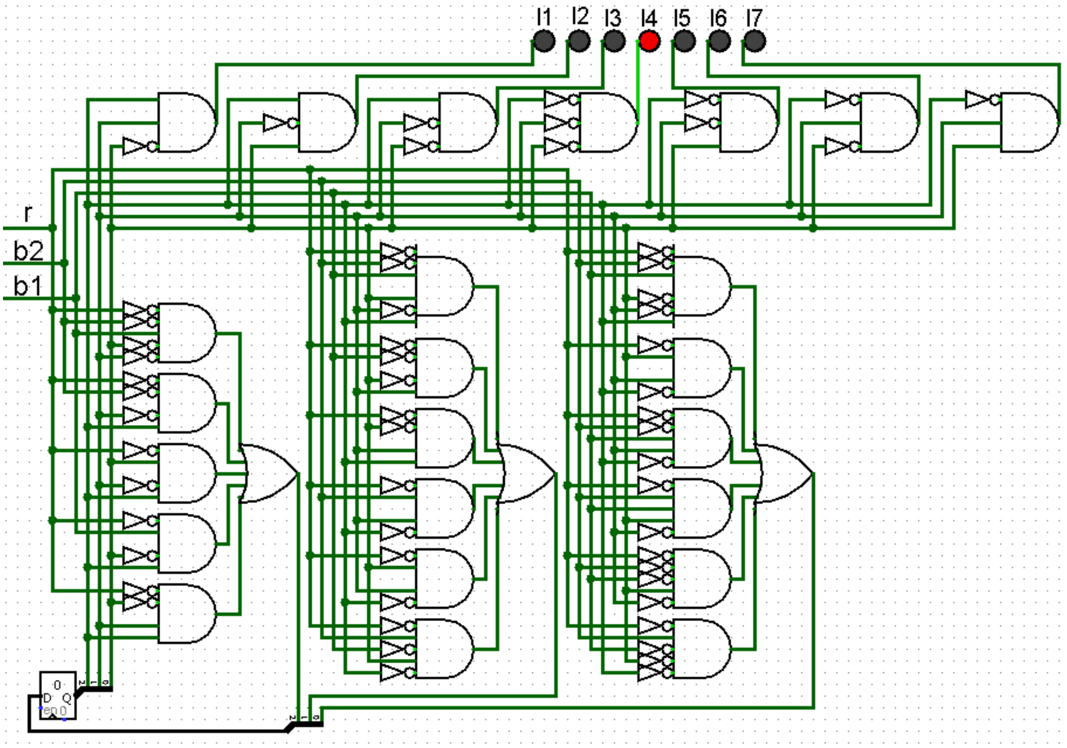
n0 = s2’s0’b1’b2r’ + s1’s0b1’b2’r’ + s1’s0b1b2r’ + s2’s1s0’b1b2’r’ + s2’s1s0r’ + s2s1’s0’b1b2’r’

n0 = s2’s0’b1’b2r’ + s1’s0b1’b2’r’ + s1’s0b1b2r’ + s2’s1r’(s0’b1b2’ + s0) + s2s1’s0’b1b2’r’

n0 = s2’s0’b1’b2r’ + s1’s0b1’b2’r’ + s1’s0b1b2r’ + s2’s1b1b2’r’ + s2’s1s0r’ + s2s1’s0’b1b2’r’

- l1 = s2s1s0’
- l2 = s2s1’s0
- l3 = s2s1’s0’
- l4 = s2’s1’s0’
- l5 = s2’s1’s0
- l6 = s2’s1s0’
- l7 = s2’s1s0

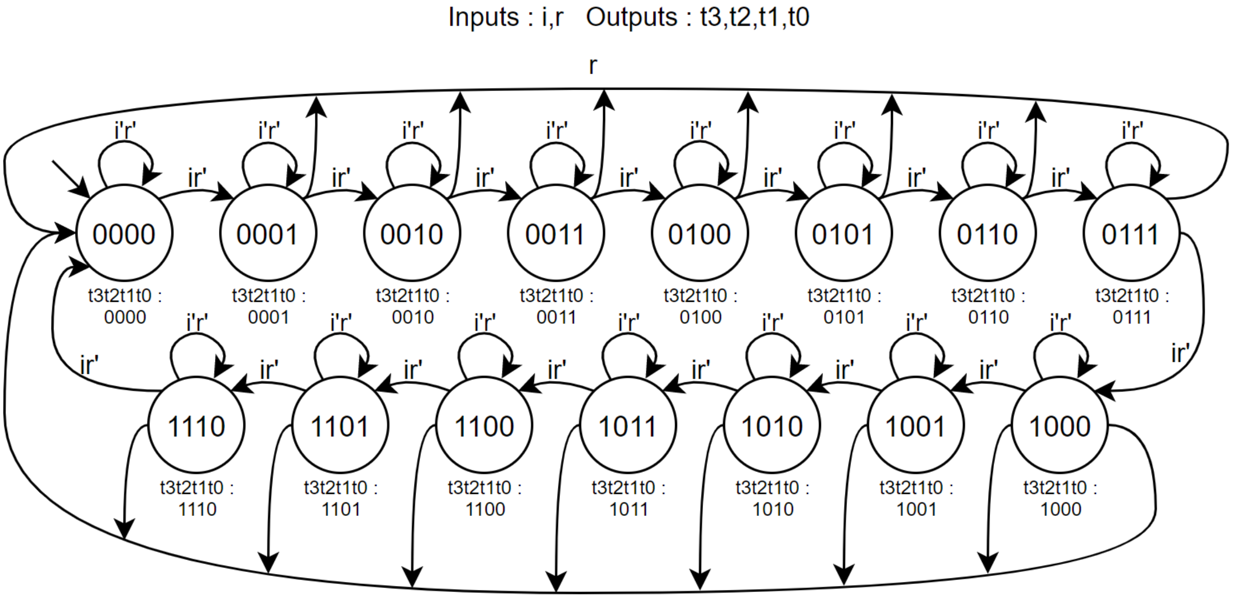
Circuit on Logisim



Score FSM

State Diagram

Note : Input i always output 1 clock timelength 1 output. Input synchronizer provides this.



Truth Table

Score Combinational Logic									
Inputs						Outputs			
p3	**p2**	**p1**	**p0**	**i**	**r**	**t3**	**t2**	**t1**	**t0**
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	1	0	0	0	1	1
0	0	1	1	0	0	0	0	1	1
0	0	1	1	1	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0
0	1	0	0	1	0	0	1	0	1
0	1	0	1	0	0	0	1	0	1
0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	0	0	1	1	0
0	1	1	0	1	0	0	1	1	1
0	1	1	1	0	0	0	1	1	1
0	1	1	1	1	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	1	0	0	1
1	0	0	1	0	0	1	0	0	1
1	0	0	1	1	0	1	0	1	0
1	0	1	0	0	0	1	0	1	0
1	0	1	0	1	0	1	0	1	1
1	0	1	1	0	0	1	0	1	1
1	0	1	1	1	0	1	1	0	0
1	1	0	0	0	0	1	1	0	0
1	1	0	0	1	0	1	1	0	1
1	1	0	1	0	0	1	1	0	1
1	1	0	1	1	0	1	1	1	0
1	1	1	0	0	0	1	1	1	0
1	1	1	0	1	0	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	0	0
X	X	X	X	X	1	0	0	0	0

Boolean Expressions

- t3

When t3 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	0
1	1	1	1	0	0

After some logical algebra

When t3 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
0	1	1	1	1	0
1	X	X	0	X	0
1	X	X	1	0	0
1	0	X	1	1	0
1	1	0	1	1	0

t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3p0i’r2’ + p3p2’p0ir2’ + p3p2p1’p0ir2’
t3 = p3’p2p1p0ir2’ + p3r2’(p0’ + p2’p0i) + p3p0r2’(i’ + p2p1’i)
t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3p2’ir2’ + p3p0i’r2’ + p3p2p1’p0r2’
t3 = p3’p2p1p0ir2’ + p3r2’(p0’ + p2p1’p0) + p3p2’ir2’ + p3p0i’r2’
t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3p2p1’r2’ + p3p2’ir2’ + p3p0i’r2’
t3 = p3’p2p1p0ir2’ + p3r2’(p0’ + p0i’) + p3p2p1’r2’ + p3p2’ir2’
t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3i’r2’ + p3p2p1’r2’ + p3p2’ir2’
t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3r2’(i’ + p2’i) + p3p2p1’r2’
t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3i’r2’ + p3p2’r2’ + p3p2p1’r2’
t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3i’r2’ + p3r2’(p2’ + p2p1’)
t3 = p3’p2p1p0ir2’ + p3p0’r2’ + p3i’r2’ + p3p2’r2’ + p3p1’r2’

- t2

When t2 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0

After some logical algebra

When t2 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
X	0	1	1	1	0
X	1	X	0	X	0
X	1	X	1	0	0
X	1	0	1	1	0

1	1	1	0	0	0
1	1	1	0	1	0
1	1	1	1	0	0

t2 = p2’p1p0ir2’ + p2p0’r2’ + p2p0i’r2’ + p2p1’p0ir2’

t2 = p2’p1p0ir2’ + p2p0’r2’ + p2p0r2’(i’ + p1’i)

t2 = p2’p1p0ir2’ + p2p0’r2’ + p2p0i’r2’ + p2p1’p0r2’

t2 = p2’p1p0ir2’ + p2r2’(p0’ + p0i’) + p2p1’p0r2’

t2 = p2’p1p0ir2’ + p2p0’r2’ + p2i’r2’ + p2p1’p0r2’

t2 = p2’p1p0ir2’ + p2r2’(p0’ + p1’p0) + p2i’r2’

t2 = p2’p1p0ir2’ + p2p0’r2’ + p2p1’r2’ + p2i’r2’

- t1

When t1 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	0
1	1	1	1	0	0

After some logical algebra

When t1 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
X	X	0	1	1	0
X	X	1	0	X	0
X	X	1	1	0	0

t1 = p1’p0ir2’ + p1p0’r2’ + p1p0i’r2’

t1 = p1’p0ir2’ + p1r2’(p0’ + p0i’)

t1 = p1’p0ir2’ + p1p0’r2’ + p1i’r2’

- t0

When t0 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	0	0

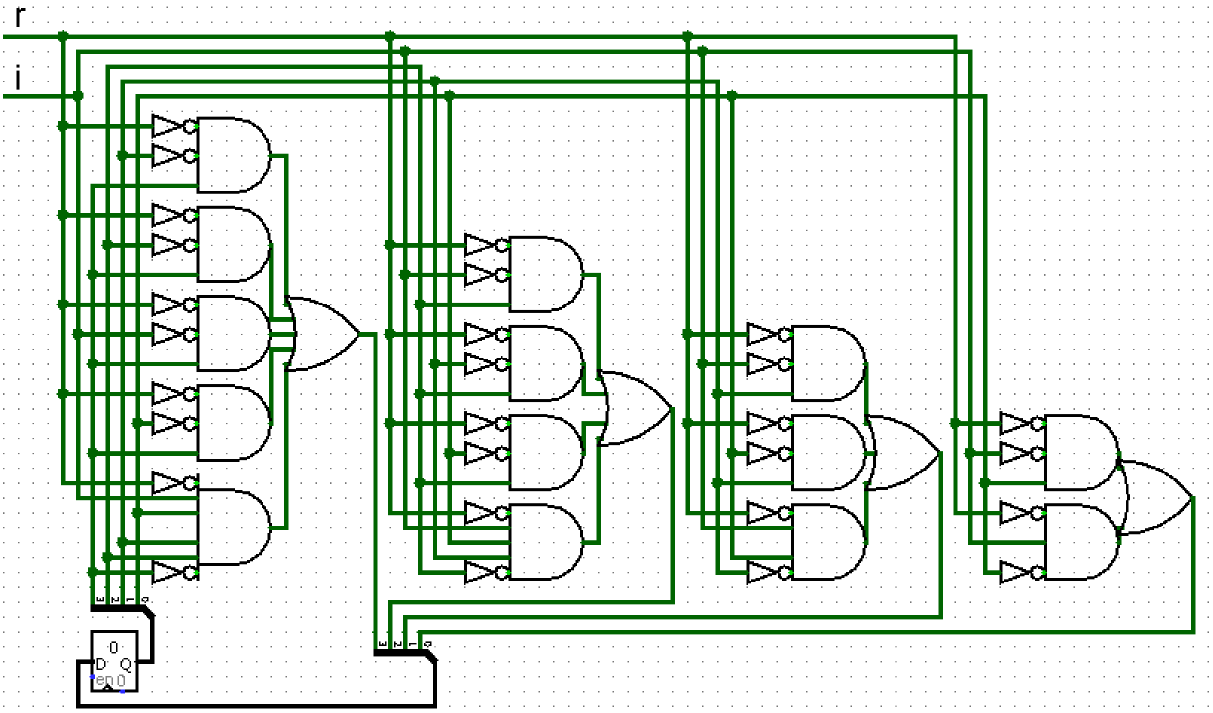
After some logical algebra

When t0 outputs is 1					
Inputs					
p3	**p2**	**p1**	**p0**	**i**	**r**
X	X	X	0	1	0
X	X	X	1	0	0

1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	1	0
1	1	1	1	0	0

t0 = p0’ir2’ + p0i’r2’

Circuit on Logisim



System Simulate Results

Scenerio	Expected Result	Actual Result	Pass/Fail
When stage is 000, just press b1 button	Successfully went next stage(stage 100 is next stage for button b1)	As expected	Pass
When stage is 000, just press b2 button	Successfully went next stage(stage 001 is next stage for button b2)	As expected	Pass
When stage is 000, just press b1 and b2 buttons or don't press any buttons	Successfully kept current stage	As expected	Pass
When stage is 001, just press b1 button	Successfully turned previous stage(stage 000 is previous stage for button b1)	As expected	Pass

When stage is 001, just press b2 button	Successfully went next stage(stage 010 is next stage for button b2)	As expected	Pass
When stage is 001, just press b1 and b2 buttons or don't press any buttons	Successfully kept current stage	As expected	Pass
When stage is 010, just press b1 button	Successfully turned previous stage(stage 001 is previous stage for button b1)	As expected	Pass
When stage is 010, just press b2 button	Successfully went next stage(stage 011 is next stage for button b2)	As expected	Pass
When stage is 001, just press b1 and b2 buttons or don't press any buttons	Successfully kept current stage	As expected	Pass
When stage is 011, press button b1 or press button b2 or press b1 and b2 buttons	Successfully kept current stage	As expected	Pass
When stage is 100, just press button b1	Successfully went next stage(stage 101 is next stage for button b1)	As expected	Pass
When stage is 100, just press button b2	Successfully turned previous stage(stage 000 is previous stage for button b2)	As expected	Pass
When stage is 100, just press b1 and b2 buttons or don't press any buttons	Successfully kept current stage	As expected	Pass
When stage is 101, just press button b1	Successfully went next stage(stage 110 is next stage for button b1)	As expected	Pass
When stage is 101, just press button b2	Successfully turned previous stage(stage 100 is previous stage for button b2)	As expected	Pass
When stage is 101, just press b1 and b2 buttons or don't press any buttons	Successfully kept current stage	As expected	Pass

When stage is 110, press button b1 or press button b2 or press b1 and b2 buttons	Successfully kept current stage	As expected	Pass
When stage is any stage, press the r button regardless of the status of the other buttons	Successfully turned 000 stage	As expected	Pass
When stage is 011, scoreboard increased.	Succeesfully increased	As expected	Pass
When stage is 110, scoreboard increased.	Succeesfully increased	As expected	Pass
When press reset button of scoreboard	Succesfully reseted	As expected	Pass