CS/ECE/EEE/INSTR F215:Digital Design

Lecture 33: ASM_2

Thu, 02 Dec 2021

BITS Pilani
Hyderabad Campus

Dr. R. N. Ponnalagu, EEE

"Today could be one day

closer to success

A little progress each day adds

up to big results".

Example

- Design a Digital system having two FFs E and F, one 4-bit binary counter A (individual FFs :A3A2A1A0).
- Initial state is the reset state: S_idle (state reached by application of reset_b)
- A Start signal initiates the operation by clearing counter A and flip-flop F.
- Counter incremented by one starting from next clock pulse, continues to increment until operations are such that it stops.
- Counter bits A2 and A3 determine sequence of operations
  - ✓ If A2 = 0, E is cleared to 0 and count continues
  - ✓ If A2 = 1, E set to 1; then if A3=0, count continues,
  - ✓ but if A3 = 1, F is set to 1 on next clock pulse and system stops counting.
- If Start = 0 system remains in initial state, but if Start = 1 cycle repeats.

- When no operations the system is in initial state *S_idle* awaiting for *Start* signal

- When input *Start* = 1, state changes to *S_1* and the counter A and flip-flop F are cleared. The register operations occur unconditionally (Moore type).

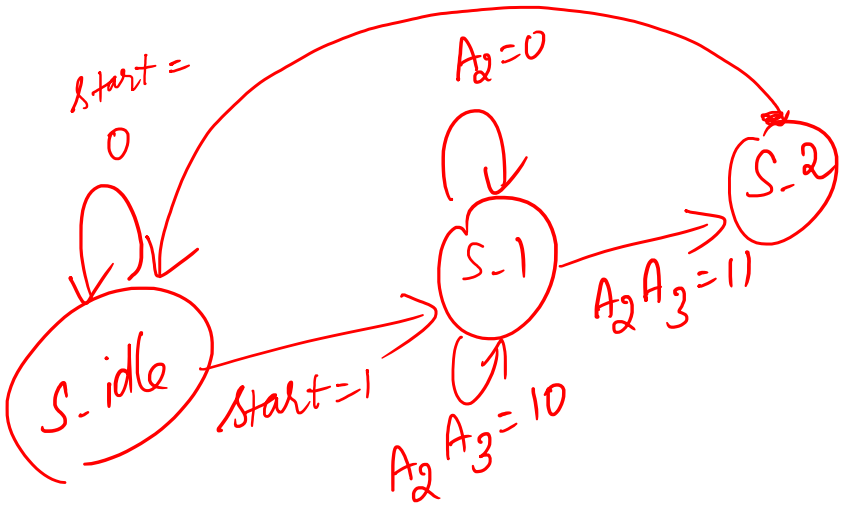
- Register A is incremented at every clock edge while machine is state *S_1*.

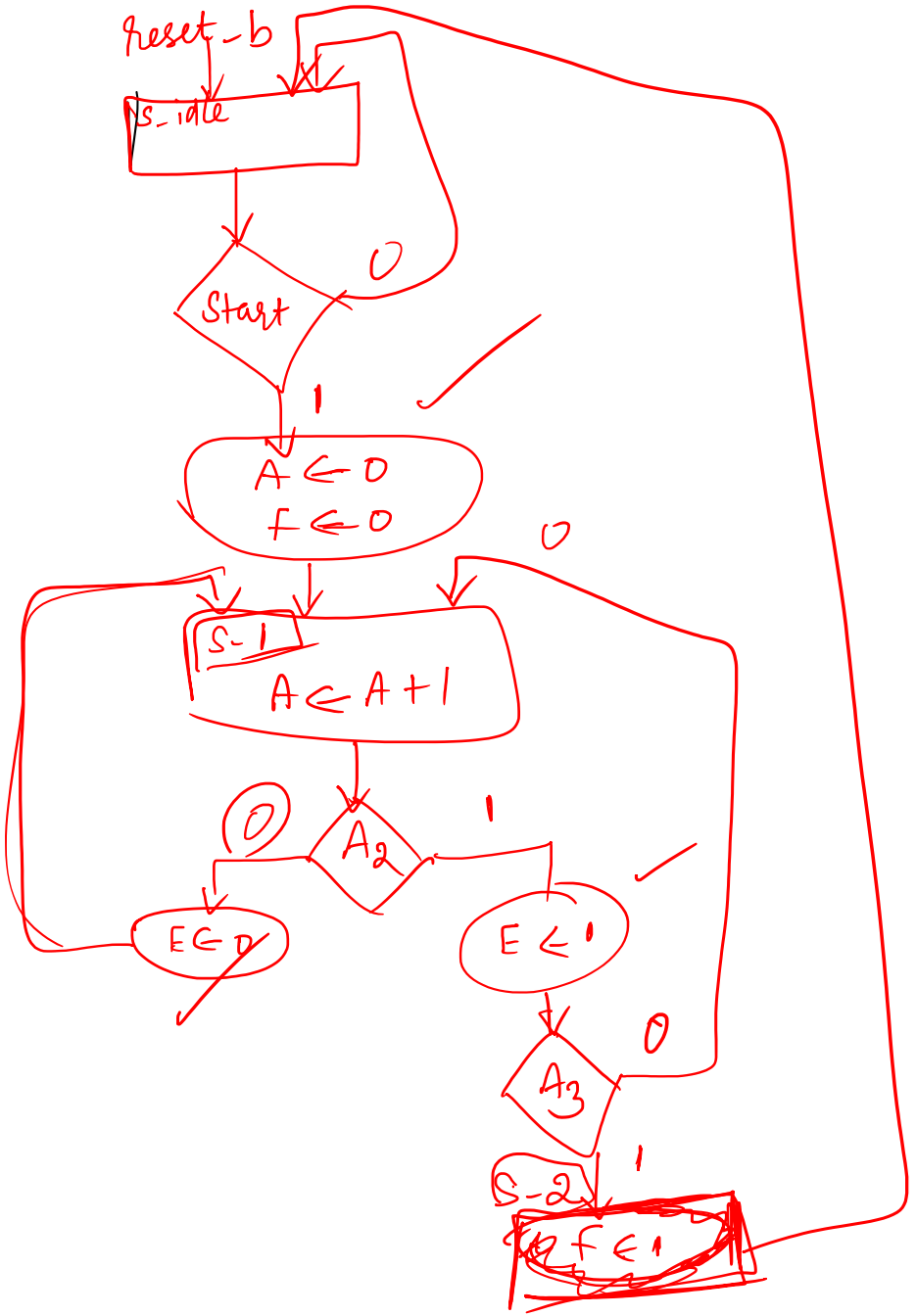
- In state *S_1* when the counter is incremented with every clock pulse and at the same time, one of three operations occur during clock transition either

  - ✓ E is cleared and control stays in state *S_1* (A2 = 0); or
  - ✓ E is set and control stays in state *S_1* (A2A3 = 10); or
  - ✓ E is set and control goes to state *S_2* (A2A3 = 11).

- Hence block with state *S_1* has two decision and two conditional boxes

- In state *S_2*, a Moore type control signal is asserted to set flipflop F and state changes to *S_idle*.


start =
0
S_idle
start=1
S_1
A2=0
A2 A3=10
S_2
A2 A3=11


reset_b
S_idle
Start
0
1
A ⟵ 0
f ⟵ 0
S_1
A ⟵ A+1
A2
0
1
E ⟵ 0
E ⟵ 1
A3
0
1
S_2
f ⟵ 1

Sequence of operations

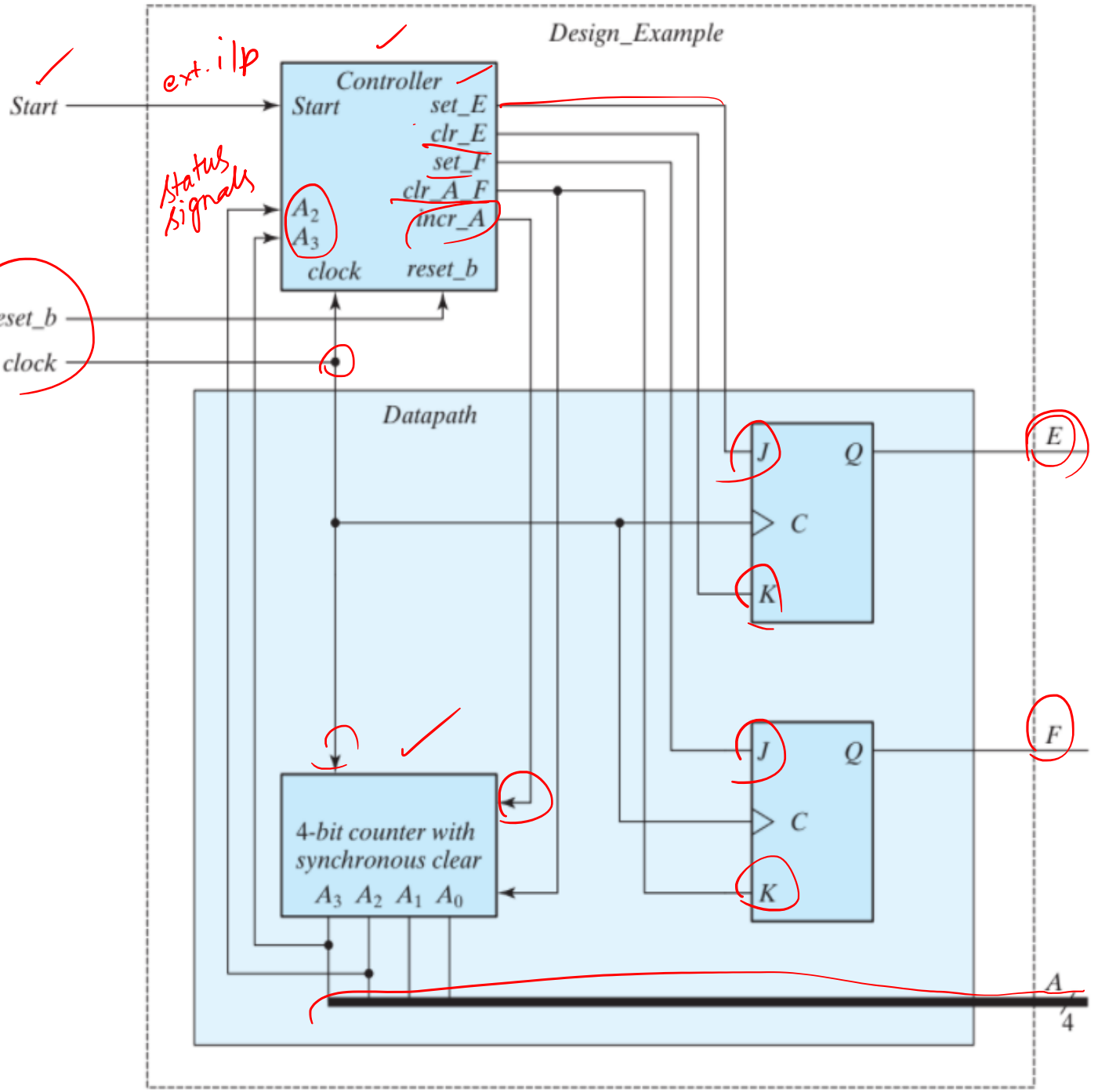
Counter				Flipflops			
A3	A2	A1	A0	E	F	Conditions	State
0	0	0	0	1	0	A2= 0, A3=0	*S_1*
0	0	0	1	0	0		
0	0	1	0	0	0		
0	0	1	1	0	0		
0	1	0	0	0	0	A2=1, A3= 0	
0	1	0	1	1	0		
0	1	1	0	1	0		
0	1	1	1	1	0		
1	0	0	0	1	0	A2=1, A3= 0	
1	0	0	1	0	0		
1	0	1	0	0	0		
1	0	1	1	0	0		
1	1	0	0	0	0	A2=1, A3= 1	
1	1	0	1	1	0		*S_2*
1	1	0	1	1	1		*S_idle*

Table 6.3

Sequence of Operations for Design Example

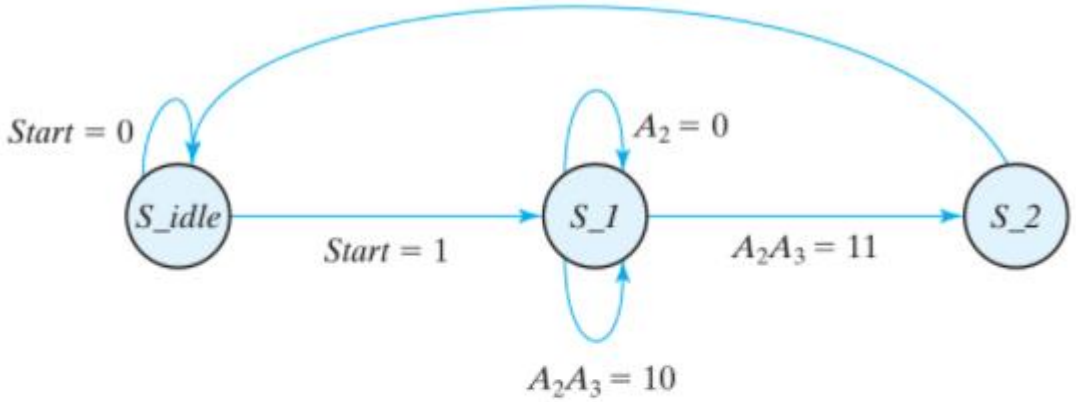
Counter				Flip-Flops			
A_3	A_2	A_1	A_0	E	F	Conditions	State
0	0	0	0	1	0	$A_2 = 0, A_3 = 0$	*S_1*
0	0	0	1	0	0		
0	0	1	0	0	0		
0	0	1	1	0	0		
0	1	0	0	0	0	$A_2 = 1, A_3 = 0$	
0	1	0	1	1	0		
0	1	1	0	1	0		
0	1	1	1	1	0		
1	0	0	0	1	0	$A_2 = 0, A_3 = 1$	
1	0	0	1	0	0		
1	0	1	0	0	0		
1	0	1	1	0	0		
1	1	0	0	0	0	$A_2 = 1, A_3 = 1$	
1	1	0	1	1	0		*S_2*
1	1	0	1	1	1		*S_idle*

- The data path unit consists of a **four-bit binary counter and two JK flip-flops.**
- The counter is incremented with every clock pulse when the controller state is S_1 .
- It is cleared only when control is at state S_idle and Start is equal to 1.
- The logic for the signal clr_A_F will be included in the controller and requires an AND gate to guarantee that both conditions are present.
- Similarly, we can anticipate that the controller will use AND gates to form signals set_E and clr_E .
- Depending on whether the controller is in state S_1 and whether A2 is asserted, set_F controls flip-flop F and is asserted unconditionally during state S_2 .
- Note that all flip-flops and registers, including the flip-flops in the control unit, use a common clock.



Design of logic circuit of the controller

State diagram



(a)

S_idle	*S_1, clr_A_F:*	$A \leftarrow 0, F \leftarrow 0$
S_1	*S_1, incr_A:*	$A \leftarrow A + 1$
	if ($A_2 = 1$) then *set_E:*	$E \leftarrow 1$
	if ($A_2 = 0$) then *clr_E:*	$E \leftarrow 0$
S_2	*S_idle, set_F:*	$F \leftarrow 1$

State Table

Present-State Symbol	Present State G_1	Present State G_0	Inputs Start	Inputs A_2	Inputs A_3	Next State G_1	Next State G_0	Outputs set_E	Outputs clr_E	Outputs set_F	Outputs clr_A_F	Outputs incr_A
S_idle	0	0	0	X	X	0	0	0	0	0	0	0
S_idle	0	0	1	X	X	0	1	0	0	0	1	0
S_1	0	1	X	0	X	0	1	0	1	0	0	1
S_1	0	1	X	1	0	0	1	1	0	0	0	1
S_1	0	1	X	1	1	1	1	1	0	0	0	1
S_2	1	1	X	X	X	0	0	0	0	1	0	0

$$D_{G1} = S\_1 A_2 A_3$$

$$D_{G0} = Start\, S\_idle + S\_1$$

$$set\_E = S\_1 A_2$$
$$clr\_E = S\_1 A_2'$$
$$set\_F = S\_2$$
$$clr\_A\_F = Start\, S\_idle$$
$$incr\_A = S\_1$$

S_idle → 00
S_1 → 01
S_2 → 11

S_idle | S_1 | S_2 | D_{G1} | D_{G0}
0 0 | 0 0 | 0 0 | 0 | 0
0 0 | 0 0 | 0 0 | 0 | 1
0 0 | 0 1 | 0 0 | 1 | 1
0 0 | 1 0 | 0 0 | 1 | 1
0 0 | 1 1 | 0 0 | 1 | 1
0 0 | 0 0 | 1 1 | 0 | 0

$S_2 = G_1 G_0$

$S_{idle} = \overline{G_1 G_0}$

$= \overline{G_0}$

Set_E
clr_E
set_f
clr_A_f
incr_A

$S\_1 A_2 = \overline{G_1} G_0 A_2$

$S\_1 \overline{A_2} = \overline{G_1} G_0 \overline{A_2}$

$S\_2 = G_1 G_0$

$S_{idle}\, Start =$

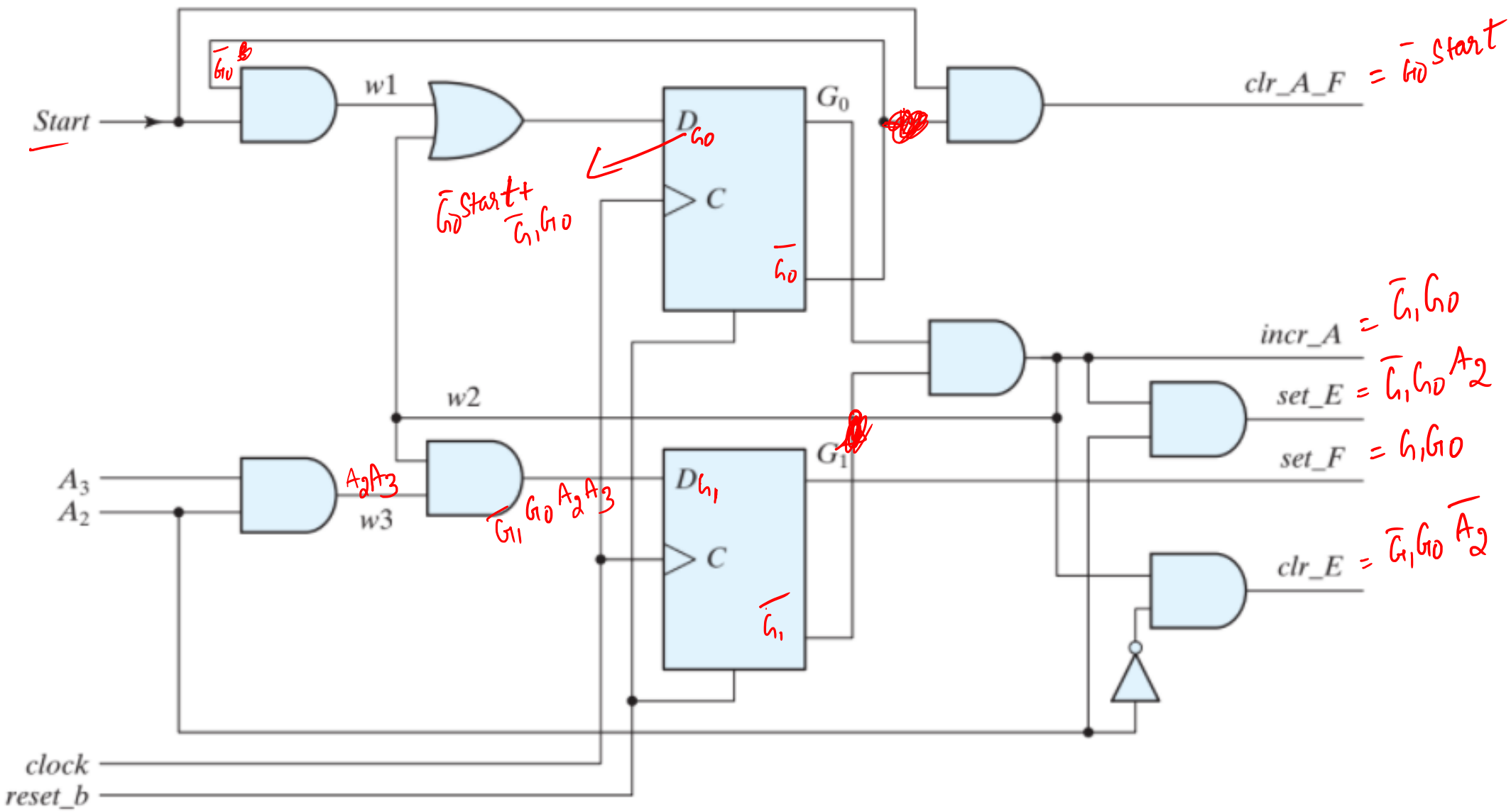
$= \overline{G_1} \overline{G_0}$

Set

$D_{G1} = S\_1 A_2 A_3$

$D_{G0} = S_{idle}\, Start + S\_1$

Logic circuit of the controller



BINARY MULTIPLIER

To Multiply two unsigned binary numbers.

Sequential Multiplier

Uses One adder and a shift register.

Less hardware but takes more clock cycles to complete the operation

The process consists of successively adding and shifting copies of the multiplicand.

The product obtained from the multiplication of two binary numbers of n bits each can have up to 2 n bits.

multiply the two binary numbers 10111 and 10011:

23	10111	multiplicand
19	10011	multiplier
	10111	
	10111	
	00000	
	00000	
	10111	
437	110110101	product

Block diagram