

به نام خدا

تمرین سوم فاز اول

دکتر مدرسی

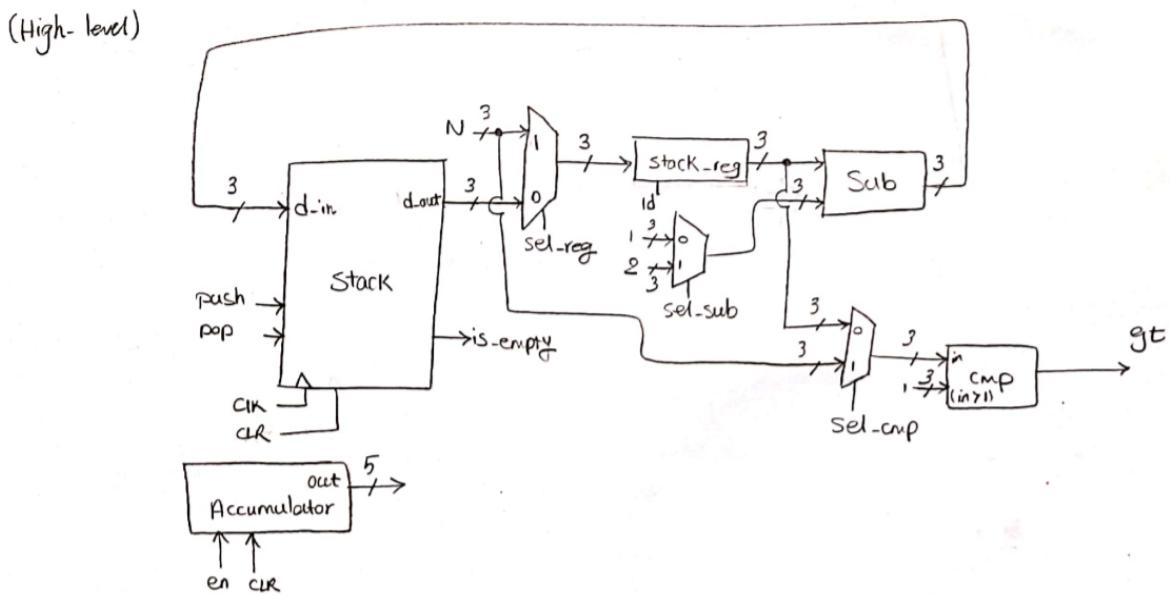
اعضای گروه:

پریا خوشتاب 810198387

پرنیان فاضل 810198516

❖ Datapath

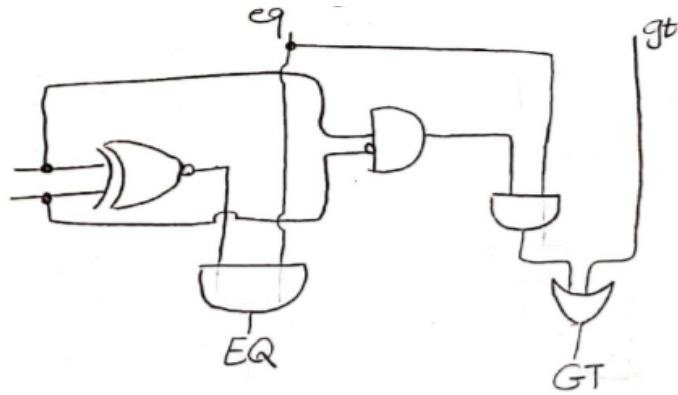
ابتدا مسیر داده را به صورت high-level پیاده سازی میکنیم:



در این مسیر داده ما از یک استک استفاده کردیم که سیگنال های ورودی push و pop داشته و سیگنال خروجی is_empty دارد. در هر مرحله مقدار بالای استک را با استفاده از سیگنال pop در رजیستر stack_reg ذخیره کرده و با استفاده از مالتی پلکسر، مقدار 1 یا 2 را از خروجی رجیستر کم میکنیم و آن را دوباره در استک پوش میکنیم تا پارامتر تابع های بازگشتی جدید در استک ایجاد شود و تا زمانی این کار را ادامه میدهیم که به شرط بازگشت تابع بازگشتی برسیم. مقدار نهایی در Accumulator ذخیره میشود.

حال هر کدام از component های بالا در سطح گیت با استفاده از سلول های منطقی داده شده پیاده سازی میکنیم:

Comparator ●



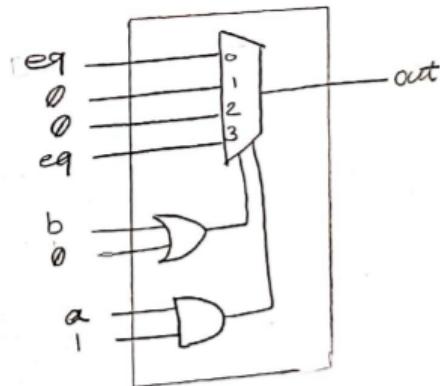
$$GT = gt \mid (eq \ \& \ a \ \& \ \sim b)$$

$$EQ = eq \ \& \ (\sim a \ \wedge \ b)$$

یک بیتی را با استفاده از یک سلول C2 به صورت زیر می سازیم:

با استفاده از قانون Shannon Expansion Theorem روی متغیرهای a و b داریم:

$$EQ = a \cdot b \cdot (eq) + a \cdot b' \cdot (0) + a' \cdot b \cdot (0) + a' \cdot b' \cdot (eq)$$



یک بیتی را با استفاده از دو سلول C1 به صورت زیر پیاده سازی می کنیم:

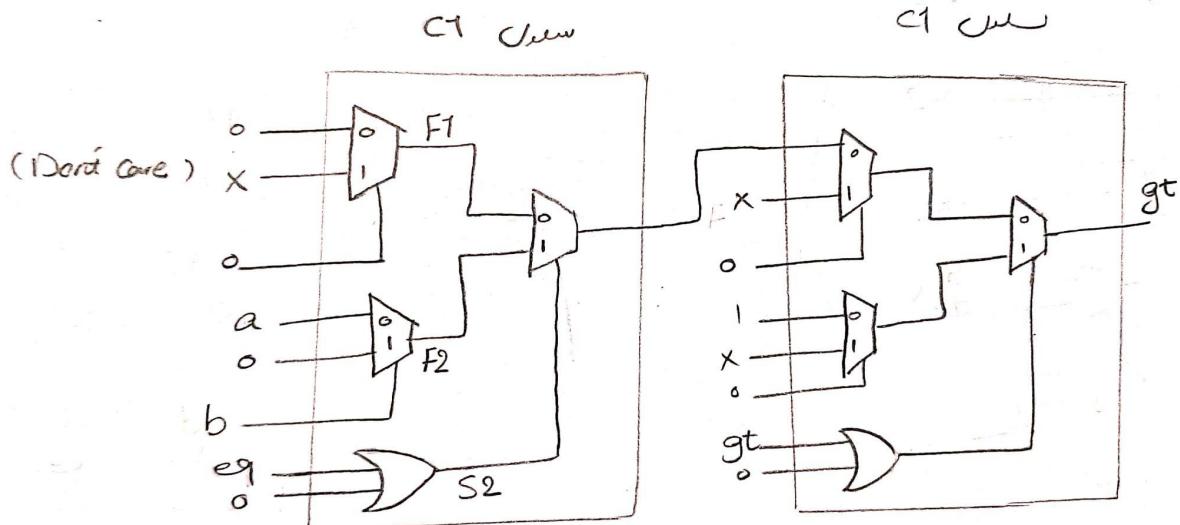
با استفاده از قانون Shannon Expansion Theorem روی متغیر gt داریم:

$$GT = gt' \cdot (eq \cdot a \cdot b') + gt \cdot (1)$$

با استفاده از قانون Shannon Expansion Theorem روی متغیر eq در عبارت $a \cdot b$

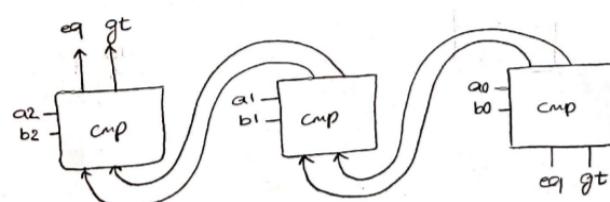
داریم:

$$eq \cdot a \cdot b = eq' \cdot (0) + eq \cdot (a \cdot b')$$



با توجه به این که طبق صورت سوال، ورودی های ما 3 بیتی هستند، باید عمل cascading را مطابق شکل زیر انجام دهیم تا از وصل کردن سلول های 1 بیتی، یک مقایسه کننده 3 بیتی

بسازیم:



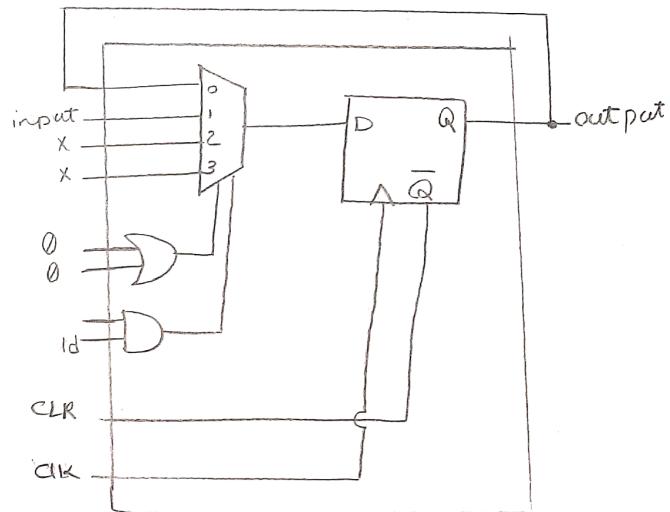
Register •

رجیستر 1 بیتی را با استفاده از یک سلول S2 پیاده سازی می‌کنیم:

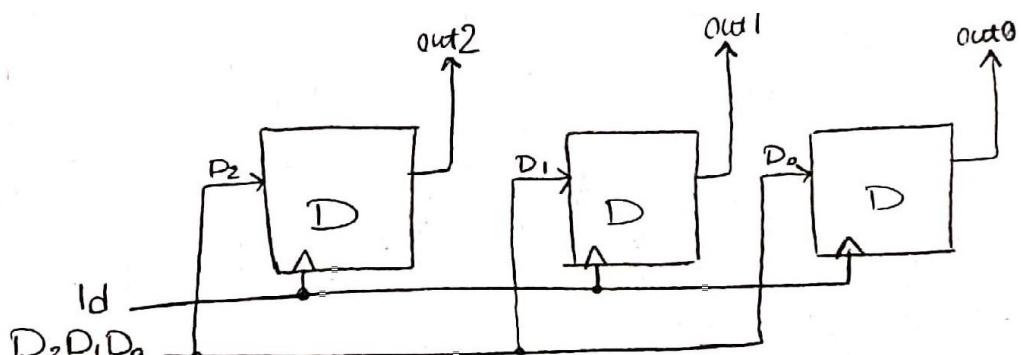
با استفاده از قانون Shannon Expansion Theorem روی متغیر ld داریم:

$$Q+ = ld' \cdot (Q) + ld \cdot (\text{input})$$

 Register :



حال از cascade کردن 3 تا رجیستر 1 بیتی، یک رجیستر 3 بیتی می‌سازیم:



Multiplexer •

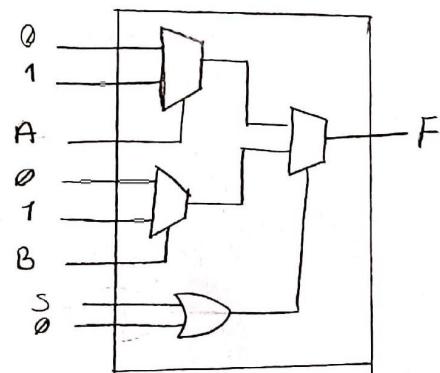
یک بیتی را با استفاده از یک سلول C1 به صورت زیر پیاده سازی می کنیم:

با استفاده از قانون Shannon Expansion Theorem روی متغیر S داریم:

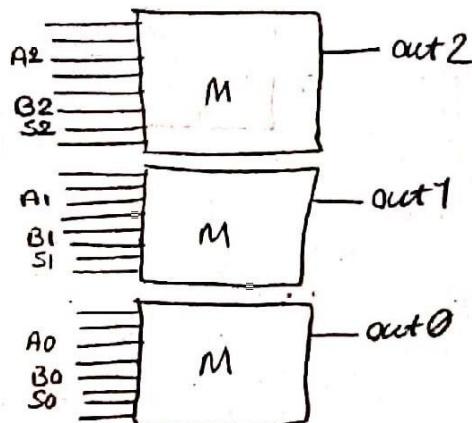
$$F = S' \cdot (A) + S \cdot (B)$$

 Multiplexer: \rightarrow
(2:1)

سلول C1

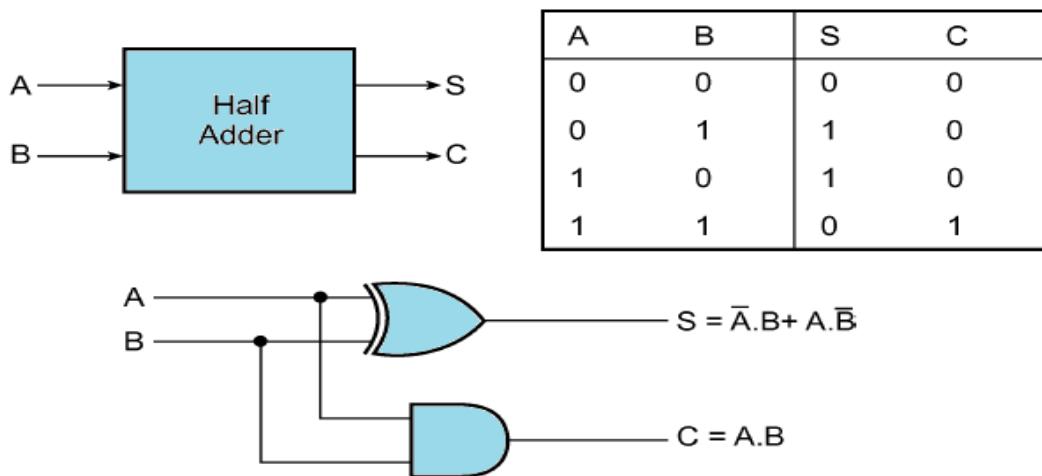


حال سلول بالا M نامیده و آن را با cascade کردن به 3 mux 3 بیتی تبدیل میکنیم:



Accumulator •

برای ساخت half adder از accumulator می کنیم.



S را با استفاده از یک سلول S2 و C را با استفاده از یک سلول C1 پیاده سازی می کنیم:

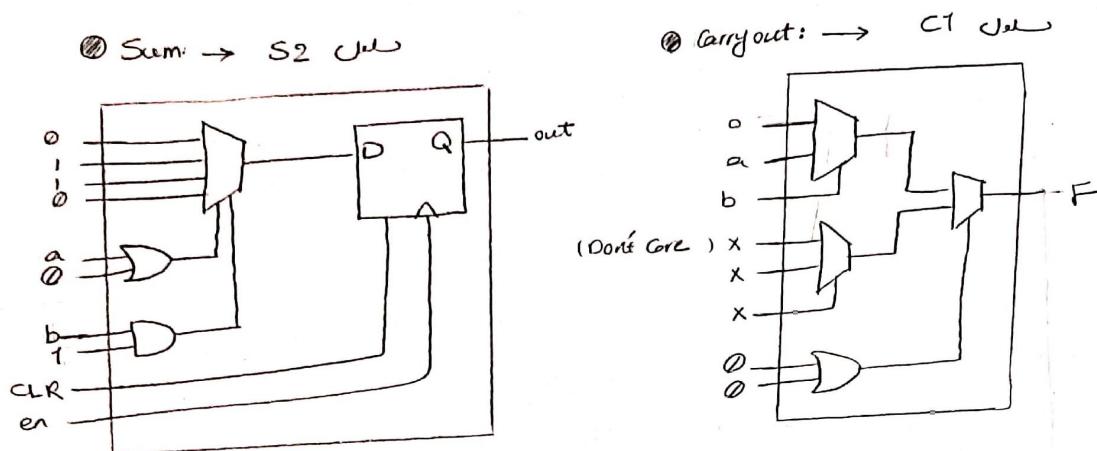
با استفاده از قانون Shannon Expansion Theorem روی متغیرهای a و b داریم:

$$S = a \cdot b \cdot (0) + a' \cdot b \cdot (1) + a \cdot b' \cdot (1) + a' \cdot b' \cdot (0)$$

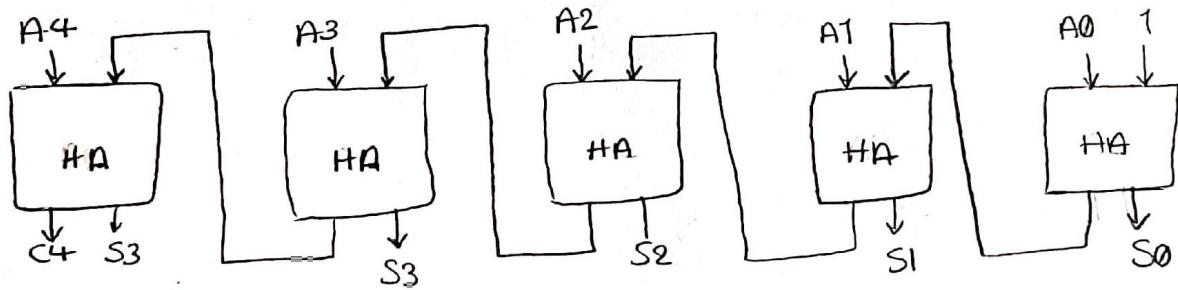
با استفاده از قانون Shannon Expansion Theorem روی متغیرهای b داریم:

$$C = b \cdot (a) + b' \cdot (0)$$

Accumulator:

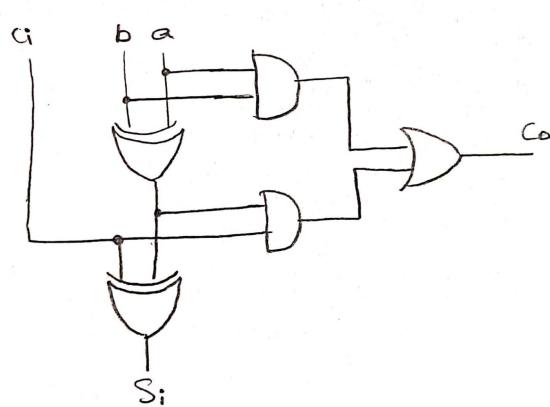


از cascade half adder های یک بیتی، مطابق شکل زیر یک accumulator پنج بیتی میسازیم. (توجه کنید که چون ورودی 3 بیتی می باشد، حداکثر به ازای $n=7$ خواهیم داشت: که یک عدد 5 بیتی می باشد، بنابراین ما به یک accumulator پنج بیتی نیاز داریم.)



Subtractor •

ابتدا با استفاده از cascade full adder یک adder می سازیم و سپس برای ساخت subtractor، ورودی اول adder را به ورودی اول subtractor میدهیم و مکمل اول (نقیض) ورودی دوم subtractor را به ورودی دوم adder می دهیم و به ورودی cin در adder عدد یک را می دهیم.



$$S = a' \cdot b' \cdot ci + a' \cdot b \cdot ci' + a \cdot b' \cdot ci' + a \cdot b \cdot ci$$

$$Co = a' \cdot b \cdot ci + a \cdot b' \cdot ci + a \cdot b \cdot ci' + a \cdot b \cdot ci$$

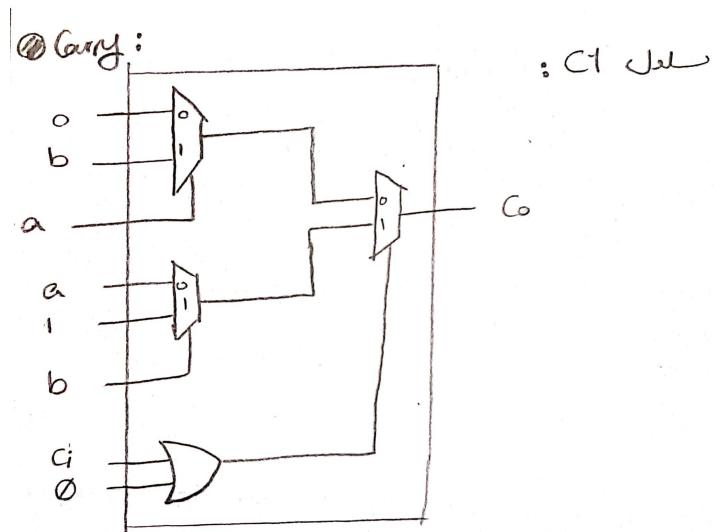
$$= a \cdot b \cdot ci' + ci(a' \cdot b + a \cdot b' + a \cdot b)$$

$$= a \cdot b \cdot ci' + ci(b + a \cdot b')$$

Co را با استفاده از یک سلول C1 پیاده سازی می‌کنیم:

با استفاده از قانون **Shannon Expansion Theorem** روی متغیر ci داریم:

$$Co = ci' \cdot (a \cdot b) + ci \cdot (b + a \cdot b')$$



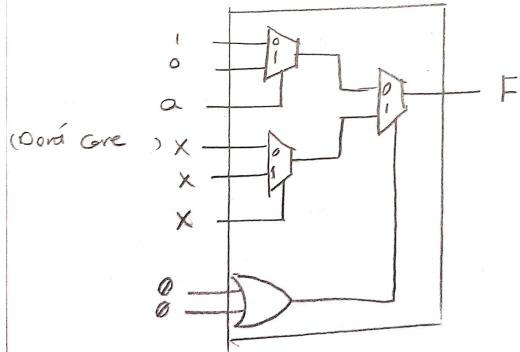
Co را با استفاده از یک سلول C1 پیاده سازی می‌کنیم:

با استفاده از قانون **Shannon Expansion Theorem** روی متغیر a داریم:

$$a' = a' \cdot (1) + a \cdot (0)$$

NOT: \rightarrow

سلول C1



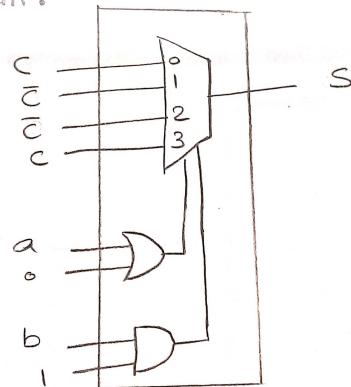
S را با استفاده از یک سلول C2 پیاده سازی می‌کنیم:

با استفاده از قانون Shannon Expansion Theorem روی متغیرهای a و b داریم:

$$S = a' \cdot b' \cdot (ci) + a' \cdot b \cdot (ci') + a \cdot b' \cdot (ci') + a \cdot b \cdot (ci)$$

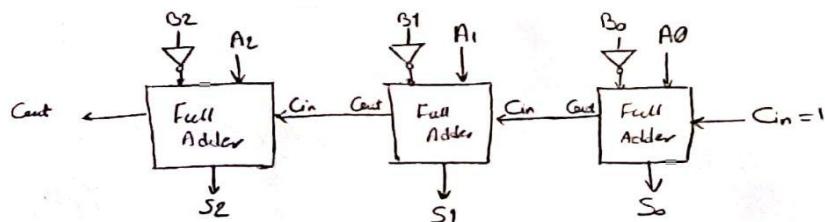
و برای ساخت ci' از ماجول Not که در بالا پیاده سازی کردیم، استفاده می‌کنیم.

Sum:

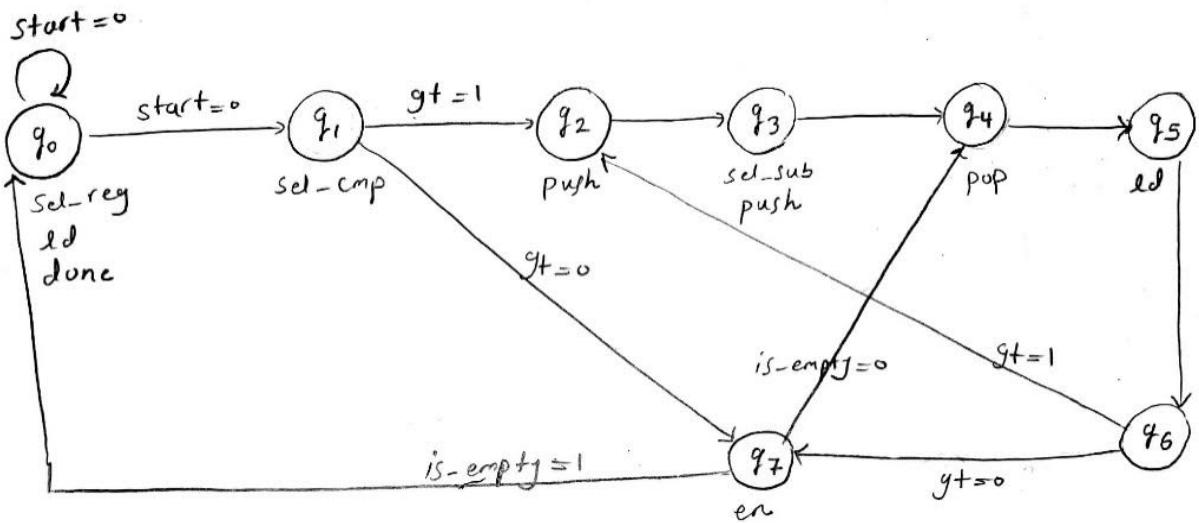


حال با استفاده از cascade کردن سه عنصر بالا که با سلول های داده شده پیاده سازی

کردیم، subtractor سه بیتی می‌سازیم:



❖ Controller



این کنترلر 8 استیت دارد. عملیات با یک کردن یک سیگنال ورودی (به نام start) شروع می‌شود. با یک شدن این سیگنال، رجیسترهای داخلی مدار ریست شده و مقدار n از ورودی خوانده می‌شود و عملیات آغاز می‌گردد. در هنگام انجام عملیات، اگر سیگنال start یک شود مدار به آن توجهی نمی‌کند و کار خود را با مقدارهای جاری ادامه میدهد. پس از اتمام عملیات، سیگنالی خروجی (به نام done) به مدت یک کلاک یک می‌شود. همانطور که دیده می‌شود در استیت های q2 و q3 پارامترهای تابع بازگشتی بعدی را در استک پوش می‌کنیم تا به شرط بازگشت تابع بازگشتی برسیم و هنگامی که به شرط تابع بازگشتی رسیدیم en را در Accumulator یک می‌کنیم و خروجی نهایی در ذخیره خواهد شد. هنگامی که استک خالی شد (سیگنال خروجی is_empty استک یک شد) کار تمام شده و سیگنال خروجی مدار done یک می‌شود.

جدول درستی کنترلر:

Inputs							outputs											
s2	s1	s0	start	gt	s_empty	s2+	s1+	s0+	sel_reg	ld	sel_cmp	sel_sub	push	pop	en	done		
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	
0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	
0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	
0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	1	
0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	
0	0	0	1	0	1	0	0	1	1	1	0	0	0	0	0	0	1	
0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	
0	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	
0	0	1	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	
0	0	1	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	
0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	
0	0	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	
0	0	1	1	0	0	1	1	1	0	0	1	0	0	0	0	0	0	
0	0	1	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	
0	0	1	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	
0	0	1	1	1	1	0	1	1	0	0	0	1	0	0	0	0	0	
0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	
0	1	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	
0	1	0	1	0	1	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	0	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	
0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	
0	1	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	1	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	
0	1	1	1	0	1	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	1	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	
0	1	1	1	1	1	1	0	1	0	0	0	0	0	1	0	0	0	
1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	
1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	
1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	
1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	
1	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	
1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	
1	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	
1	0	0	1	1	1	0	0	1	0	0	0	0	0	1	0	0	0	
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	
1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	
1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	
1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	
1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	
1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	
1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	1	0	0	
1	0	1	1	1	1	0	1	1	0	0	0	0	0	0	1	0	0	
1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	
1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	
1	1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	0	
1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	
1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	
1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0	
1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	1	0	
1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	
1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	
1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	
1	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	
1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	
1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	
1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0	
1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	1	0	
1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	

حال با توجه به جدول درستی بالا، جدول کارنوی مربوط به هر سیگنال خروجی را رسم میکنیم و با استفاده از آن، ابتدا عبارت جبری بهینه برای هر خروجی را به دست آورده و سپس هر خروجی را با استفاده از سلول های منطقی داده شده پیاده سازی می کنیم.

► دقت شود که برای سادگی، متغیرهای ورودی جداول کارنومپ را با متغیرهای زیر

جایگزین می کنیم:

► $S_2 \sim A$

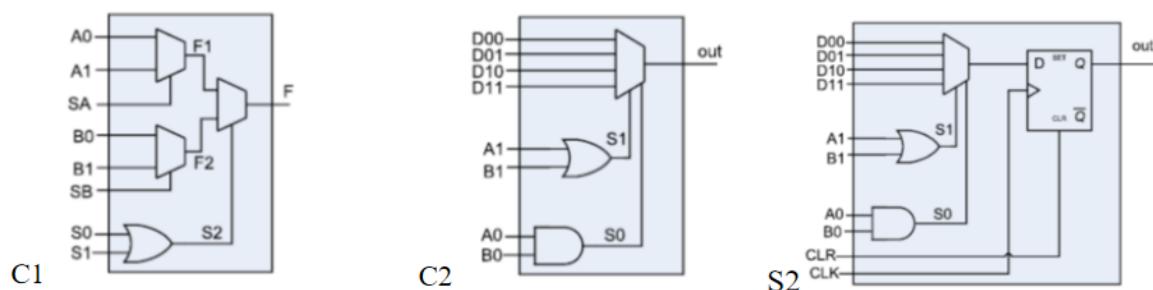
► $S_1 \sim B$

► $S_0 \sim C$

► Start $\sim D$

► $G_t \sim E$

► $Is_empty \sim F$



- S2+

	$\bar{D} \cdot \bar{E} \cdot \bar{F}$	$\bar{D} \cdot \bar{E} \cdot F$	$\bar{D} \cdot E \cdot F$	$\bar{D} \cdot E \cdot \bar{F}$	$D \cdot \bar{E} \cdot \bar{F}$	$D \cdot \bar{E} \cdot F$	$D \cdot E \cdot F$	$D \cdot E \cdot \bar{F}$
$\bar{A} \cdot \bar{B} \cdot \bar{C}$	0	0	0	0	0	0	0	0
$\bar{A} \cdot \bar{B} \cdot C$	1	1	0	0	1	1	0	0
$\bar{A} \cdot B \cdot C$	1	1	1	1	1	1	1	1
$A \cdot \bar{B} \cdot \bar{C}$	0	0	0	0	0	0	0	0
$A \cdot \bar{B} \cdot C$	1	1	1	1	1	1	1	1
$A \cdot B \cdot C$	1	0	0	1	1	0	0	1
$A \cdot B \cdot \bar{C}$	1	1	0	0	1	1	0	0

$$S2+: S2.S1' + S2'.S0.gt' + S2'.S1.S0 + S1.S0.is_empty' + S2.S0'.gt'$$

$$= S2.S1' + S1.S0 (S2' + is_empty') + gt'(S2'.S0 + S2.S0')$$

سلول S2 برای ساختن :S2+

$$D00 = 0$$

$$D01 = gt' + S1$$

$$D10 = S1' + gt'$$

$$D11 = S1' + S1.is_empty$$

$$A1 = S2$$

$$B1 = 0$$

$$A0 = S0$$

$$B0 = 1$$

$$CLR = clr$$

$$CLK = clk$$

سلول C1 برای ساختن D01 :

$$A0 = 1$$

A1 = x

SA = 0

B0 = S1

B1 = x

SB = 0

S0 = gt

S1 = 0

سلول C1 برای ساختن :D10

A0 = 1

A1 = x

SA = 0

B0 = 1

B1 = 0

SB = S1

S0 = gt

S1 = 0

سلول C1 برای ساختن :D11

A0 = 1

A1 = x

SA = 0

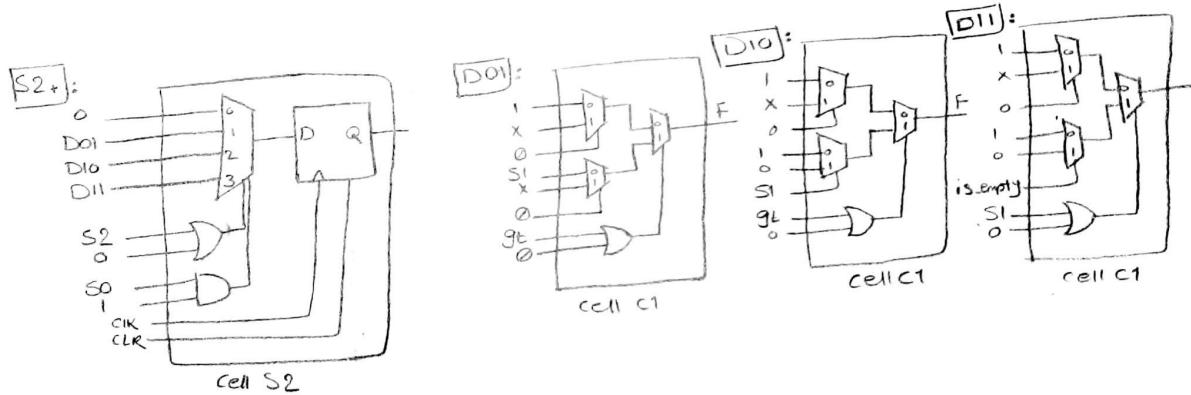
B0 = 1

B1 = 0

SB = is_empty

$$S_0 = S_1$$

$$S_1 = 0$$



● S1+

	$\bar{D} \cdot \bar{E} \cdot \bar{F}$	$\bar{D} \cdot \bar{E} \cdot F$	$\bar{D} \cdot E \cdot \bar{F}$	$\bar{D} \cdot E \cdot F$	$D \cdot \bar{E} \cdot \bar{F}$	$D \cdot \bar{E} \cdot F$	$D \cdot E \cdot \bar{F}$	$D \cdot E \cdot F$
$\bar{A} \cdot \bar{B} \cdot \bar{C}$	0	0	0	0	0	0	0	0
$\bar{A} \cdot \bar{B} \cdot C$	1	1	1	1	1	1	1	1
$A \cdot \bar{B} \cdot C$	0	0	0	0	0	0	0	0
$A \cdot \bar{B} \cdot \bar{C}$	1	1	1	1	1	1	1	1
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
$A \cdot B \cdot C$	0	0	0	0	0	0	0	0
$A \cdot \bar{B} \cdot C$	1	1	1	1	1	1	1	1
$A \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0	0
$A \cdot B \cdot \bar{C}$	1	1	1	1	1	1	1	1

$$S_1+ = S_1' \cdot S_0 + S_1 \cdot S_0'$$

سلول 2 برای ساختن S_1+ :

$$D_{00} = 0$$

$$D_{01} = 1$$

$$D_{10} = 1$$

$$D_{11} = 0$$

$$A_1 = S_1$$

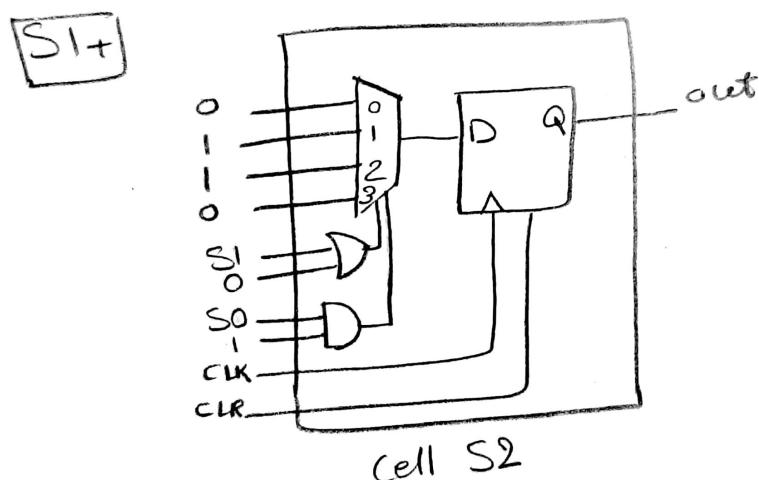
$B1 = 0$

$A0 = S0$

$B0 = 1$

$CLR = \text{clr}$

$CLK = \text{clk}$



- $S0+$

	$\bar{D}.\bar{E}.\bar{F}$	$\bar{D}.\bar{E}.F$	$\bar{D}.E.F$	$\bar{D}.E.\bar{F}$	$D.\bar{E}.\bar{F}$	$D.\bar{E}.F$	$D.E.F$	$D.E.\bar{F}$
$\bar{A}.\bar{B}.\bar{C}$	0	0	0	0	1	1	1	1
$\bar{A}.B.C$	1	1	0	0	1	1	0	0
$\bar{A}.B.C$	0	0	0	0	0	0	0	0
$\bar{A}.B.\bar{C}$	1	1	1	1	1	1	1	1
<hr/>								
$A.B.C$	1	1	1	1	1	1	1	1
$A.\bar{B}.C$	0	0	0	0	0	0	0	0
$A.B.C$	0	0	0	0	0	0	0	0
$A.B.\bar{C}$	1	1	0	0	1	1	0	0

$$S0+ = S2'.S0'.\text{start} + S2'.S1.S0' + S1.S0'.\text{gt}' + S2.S1'.S0' + S2'.S1'.S0.\text{gt}'$$

$$S0+ = S2'.S0'(start + S1) + S1.S0'.gt' + S1'(S2.S0' + S2'.S0.gt')$$

سلول S2 برای ساختن : S1+

$$D00 = start + S1$$

$$D01 = S1'.gt'$$

$$D10 = S1.gt' + S1'$$

$$D11 = 0$$

$$A1 = S2$$

$$B1 = 0$$

$$A0 = S0$$

$$B0 = 1$$

$$CLR = clr$$

$$CLK = clk$$

سلول C1 برای ساختن : D00

$$A0 = S1$$

$$A1 = x$$

$$SA = 0$$

$$B0 = 1$$

$$B1 = x$$

$$SB = 0$$

$$S0 = start$$

$$S1 = 0$$

سلول C1 برای ساختن : D01

A0 = 1

A1 = 0

SA = gt

B0 = 0

B1 = x

SB = 0

S0 = S1

S1 = 0

سلول C1 برای ساختن D10 :

A0 = 1

A1 = x

SA = 0

B0 = 1

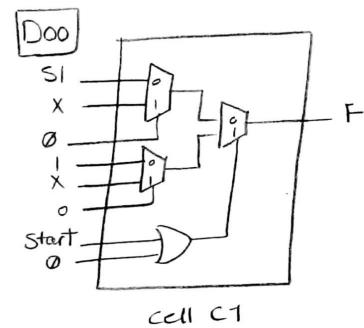
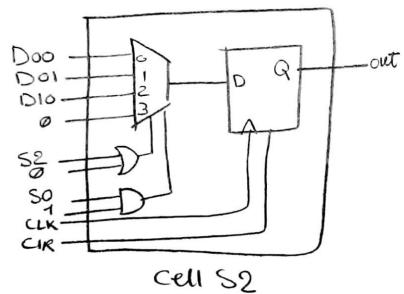
B1 = 0

SB = gt

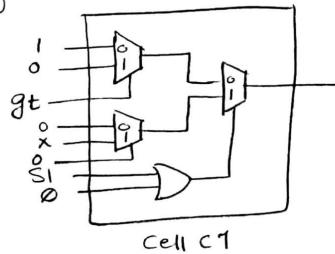
S0 = S1

S1 = 0

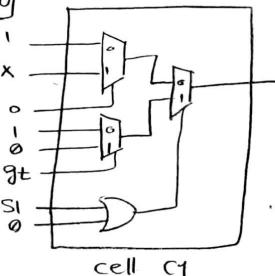
50+



D10



D10



● sel_reg

$\bar{D} \cdot \bar{E} \cdot \bar{F}$	$\bar{D} \cdot \bar{E} \cdot F$	$\bar{D} \cdot E \cdot \bar{F}$	$\bar{D} \cdot E \cdot F$	$D \cdot \bar{E} \cdot \bar{F}$	$D \cdot \bar{E} \cdot F$	$D \cdot E \cdot \bar{F}$	$D \cdot E \cdot F$
$\bar{A} \cdot \bar{B} \cdot \bar{C}$	1	1	1	1	1	1	1
$\bar{A} \cdot \bar{B} \cdot C$	0	0	0	0	0	0	0
$\bar{A} \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0
$\bar{A} \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
$A \cdot \bar{B} \cdot \bar{C}$	0	0	0	0	0	0	0
$A \cdot \bar{B} \cdot C$	0	0	0	0	0	0	0
$A \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0
$A \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0

$$Sel_reg = S2' \cdot S1' \cdot S0'$$

: Sel_reg برای ساختن C1 سلول

$$A0 = S0'$$

$$A1 = 0$$

$$SA = S1$$

$B_0 = 0$

$B_1 = 0$

$S_B = S_1$

$S_0 = S_2$

$S_1 = 0$

سلول C1 برای ساختن ' S_0'

$A_0 = 1$

$A_1 = 0$

$S_A = S_0$

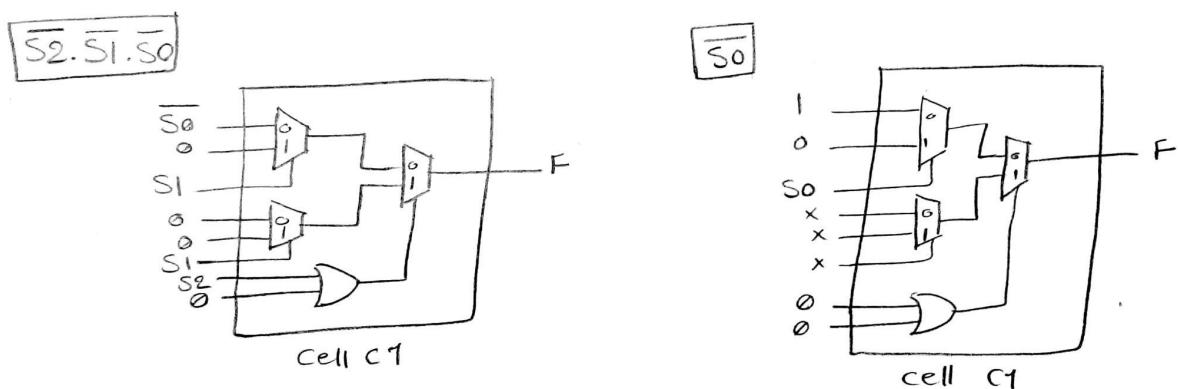
$B_0 = x$

$B_1 = x$

$S_B = x$

$S_0 = 0$

$S_1 = 0$



● 1d

	$\bar{D}.\bar{E}.\bar{F}$	$\bar{D}.\bar{E}.F$	$\bar{D}.E.F$	$\bar{D}.E.\bar{F}$	$D.\bar{E}.\bar{F}$	$D.\bar{E}.F$	$D.E.F$	$D.E.\bar{F}$
$\bar{A}.\bar{B}.\bar{C}$	1	1	1	1	1	1	1	1
$\bar{A}.\bar{B}.C$	0	0	0	0	0	0	0	0
$\bar{A}.B.C$	0	0	0	0	0	0	0	0
$A.B.\bar{C}$	0	0	0	0	0	0	0	0
$A.B.C$	0	0	0	0	0	0	0	0
$A.\bar{B}.C$	1	1	1	1	1	1	1	1
$A.B.\bar{C}$	0	0	0	0	0	0	0	0
$A.B.C$	0	0	0	0	0	0	0	0

$$ld = S2'.S1'.S0' + S2.S1'.S0$$

سلول C2 برای ساختن ld

$$D00 = S0'$$

$$D01 = 0$$

$$D10 = S0$$

$$D11 = 0$$

$$A1 = S2$$

$$B1 = 0$$

$$A0 = S1$$

$$B0 = 1$$

سلول C1 برای ساختن $S0'$

$$A0 = 1$$

$$A1 = 0$$

$$SA = S0$$

$$B0 = x$$

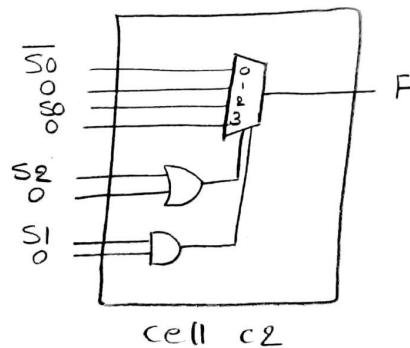
$$B1 = x$$

$$SB = x$$

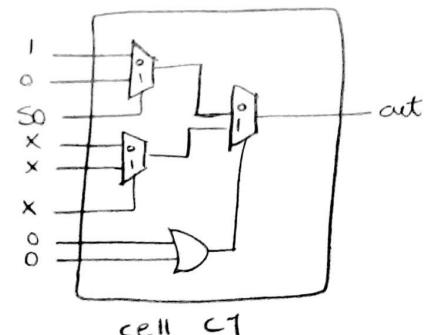
$$S0 = 0$$

$$S1 = 0$$

$|d$



$\overline{S0}$



● sel_cmp

	$\overline{D.E.F}$	$\overline{D.E.F}$	$\overline{D.E.F}$	$\overline{D.E.F}$	$D.\overline{E.F}$	$D.\overline{E.F}$	$D.E.F$	$D.E.F$
$\overline{A.B.C}$	0	0	0	0	0	0	0	0
$\overline{A}\overline{B}.C$	1	1	1	1	1	1	1	1
$A.B.C$	0	0	0	0	0	0	0	0
$A.B\overline{C}$	0	0	0	0	0	0	0	0
$A.B.C$	0	0	0	0	0	0	0	0
$A\overline{B}.C$	0	0	0	0	0	0	0	0
$A.B.C$	0	0	0	0	0	0	0	0
$A.B\overline{C}$	0	0	0	0	0	0	0	0

$$Sel_cmp = S2' \cdot S1' \cdot S0$$

:Sel_cmp برای ساختن سلول C1

$$A0 = S0$$

$$A1 = 0$$

$$SA = S1$$

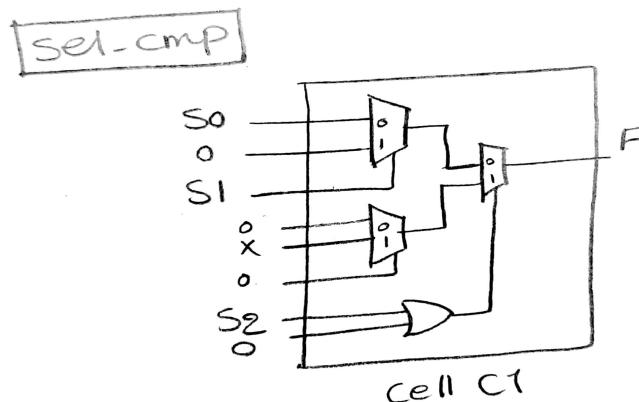
$$B_0 = 0$$

$$B_1 = x$$

$$SB = 0$$

$$S_0 = S_2$$

$$S_1 = 0$$



● sel_sub

	$\bar{D} \bar{E} \bar{F}$	$\bar{D} \bar{E} F$	$\bar{D} E \bar{F}$	$\bar{D} E \bar{F}$	$D \bar{E} \bar{F}$	$D \bar{E} F$	$D E \bar{F}$	$D E \bar{F}$
$\bar{A} \bar{B} \bar{C}$	0	0	0	0	0	0	0	0
$\bar{A} \bar{B} C$	0	0	0	0	0	0	0	0
$\bar{A} B \bar{C}$	1	1	1	1	1	1	1	1
$A \bar{B} \bar{C}$	0	0	0	0	0	0	0	0
<hr/>								
$A \bar{B} C$	0	0	0	0	0	0	0	0
$A \bar{B} \bar{C}$	0	0	0	0	0	0	0	0
$A B \bar{C}$	0	0	0	0	0	0	0	0
$A B \bar{C}$	0	0	0	0	0	0	0	0

$$Sel_{sub} = S_2' \cdot S_1 \cdot S_0$$

سلول C1 برای ساختن : Sel_sub

$$A_0 = 0$$

$$A_1 = S_0$$

$SA = S1$

$B0 = 0$

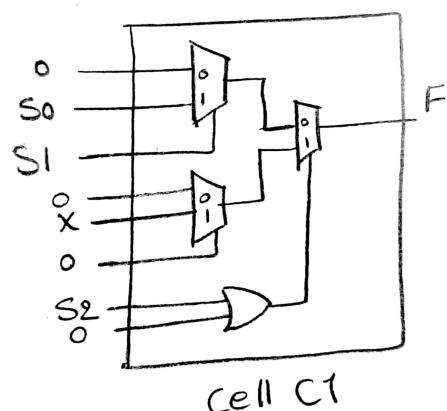
$B1 = x$

$SB = 0$

$S0 = S2$

$S1 = 0$

sel - sub



● push

	$\bar{D}\bar{E}\bar{F}$	$\bar{D}\bar{E}F$	$\bar{D}E\bar{F}$	$\bar{D}EF$	$D\bar{E}\bar{F}$	$D\bar{E}F$	$DE\bar{F}$	DEF
$\bar{A}\bar{B}\bar{C}$	0	0	0	0	0	0	0	0
$\bar{A}\bar{B}C$	0	0	0	0	0	0	0	0
$\bar{A}B\bar{C}$	1	1	1	1	1	1	1	1
$A\bar{B}\bar{C}$	1	1	1	1	1	1	1	1
$A\bar{B}C$	0	0	0	0	0	0	0	0
$A\bar{B}C$	0	0	0	0	0	0	0	0
ABC	0	0	0	0	0	0	0	0
$A\bar{B}\bar{C}$	0	0	0	0	0	0	0	0

$Push = S2'.S1$

سلول C1 برای ساختن :push

$A_0 = S_1$

$A_1 = x$

$S_A = 0$

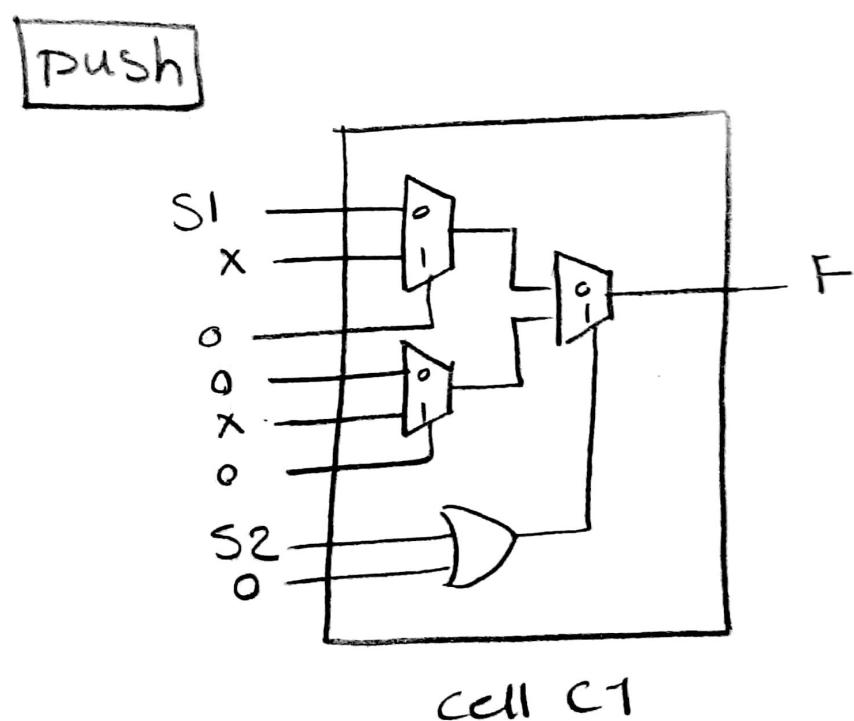
$B_0 = 0$

$B_1 = x$

$S_B = 0$

$S_0 = S_2$

$S_1 = 0$



● pop

	D.E.F							
A.B.C	0	0	0	0	0	0	0	0
A.B.C	0	0	0	0	0	0	0	0
A.B.C	0	0	0	0	0	0	0	0
A.B.C	0	0	0	0	0	0	0	0
A.B.C	1	1	1	1	1	1	1	1
A.B.C	0	0	0	0	0	0	0	0
A.B.C	0	0	0	0	0	0	0	0
A.B.C	0	0	0	0	0	0	0	0

$$\text{Pop} = S2.S1'.S0'$$

سلول C1 برای ساختن pop:

$$A0 = S2$$

$$A1 = 0$$

$$SA = S0$$

$$B0 = 0$$

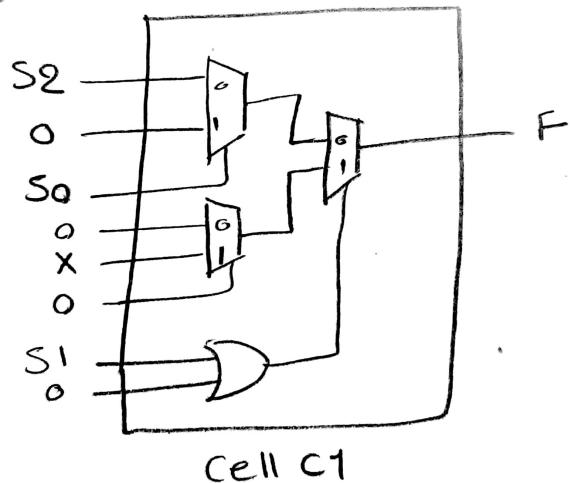
$$B1 = x$$

$$SB = 0$$

$$S0 = S1$$

$$S1 = 0$$

Pop



● en

	D̄ Ē F̄	D̄ Ē F̄	D̄ E F̄	D̄ E F̄	D̄ Ē F̄	D̄ Ē F̄	D E F̄	D E F̄
Ā B̄ C̄	0	0	0	0	0	0	0	0
Ā B̄ C	0	0	0	0	0	0	0	0
Ā B C	0	0	0	0	0	0	0	0
A B̄ C	0	0	0	0	0	0	0	0
A B C	0	0	0	0	0	0	0	0
A B C̄	0	0	0	0	0	0	0	0
A B̄ C̄	0	0	0	0	0	0	0	0
A B̄ C	1	1	1	1	1	1	1	1
A B C̄	0	0	0	0	0	0	0	0

$$En = S2 \cdot S1 \cdot S0$$

سلول C1 برای ساختن en :

$$A0 = 0$$

$$A1 = x$$

$$SA = 0$$

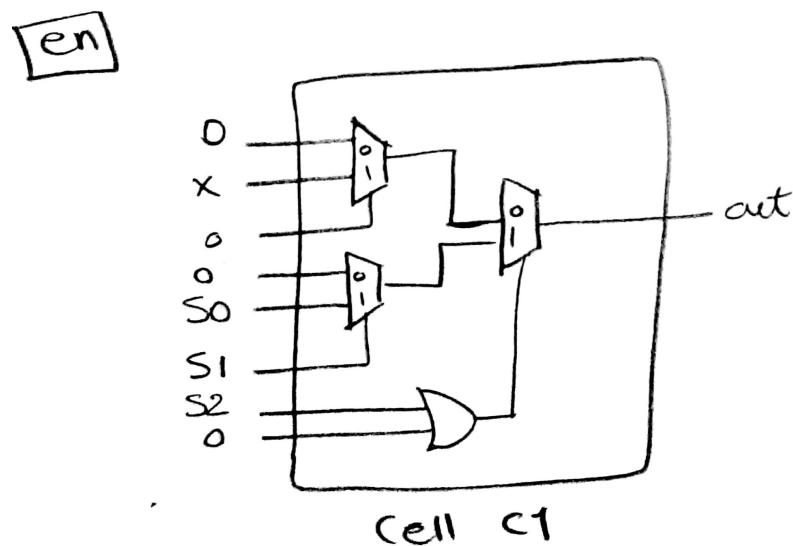
$$B0 = 0$$

$$B1 = S0$$

$$SB = S1$$

$$S0 = S2$$

$$S1 = 0$$



- done

	$\bar{D} \cdot \bar{E} \cdot \bar{F}$	$\bar{D} \cdot \bar{E} \cdot F$	$\bar{D} \cdot E \cdot \bar{F}$	$\bar{D} \cdot E \cdot F$	$D \cdot \bar{E} \cdot \bar{F}$	$D \cdot \bar{E} \cdot F$	$D \cdot E \cdot \bar{F}$	$D \cdot E \cdot F$
$\bar{A} \cdot \bar{B} \cdot \bar{C}$	1	1	1	1	1	1	1	1
$\bar{A} \cdot \bar{B} \cdot C$	0	0	0	0	0	0	0	0
$\bar{A} \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0	0
$A \cdot \bar{B} \cdot \bar{C}$	0	0	0	0	0	0	0	0
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
$A \cdot B \cdot C$	0	0	0	0	0	0	0	0
$A \cdot \bar{B} \cdot C$	0	0	0	0	0	0	0	0
$A \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0	0
$A \cdot B \cdot \bar{C}$	0	0	0	0	0	0	0	0

$$done = S2' \cdot S1' \cdot S0'$$

سلول C1 برای ساختن done:

$$A0 = S0'$$

$$A1 = 0$$

$SA = S1$

$B0 = 0$

$B1 = 0$

$SB = S1$

$S0 = S2$

$S1 = 0$

سلول C1 برای ساختن $S0'$:

$A0 = 1$

$A1 = 0$

$SA = S0$

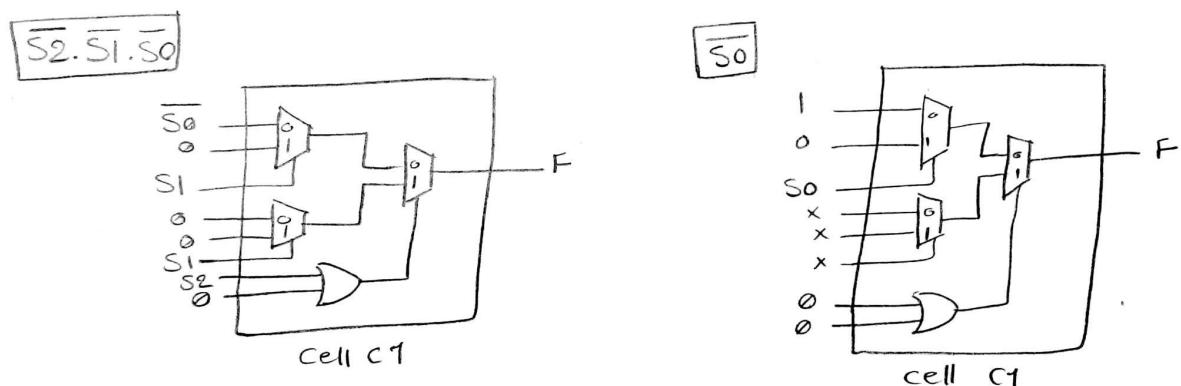
$B0 = x$

$B1 = x$

$SB = x$

$S0 = 0$

$S1 = 0$



مسیر بحرانی:

از آنجایی که استک ما سنکرون با کلاک می باشد، بنابراین استک هم به نوعی رجیستر می باشد و همچنین برای به دست آوردن مسیر بحرانی باید تاخیر بین هر دو رجیستر که یک مدار ترتیبی بین آنها وجود دارد را محاسبه کنیم و مسیر با ماکسیمم این تاخیرها را را به عنوان مسیر بحرانی در نظر بگیریم. با توجه به این توضیحات، ما مسیر بحرانی را مسیری که از رجیستر و تفریق کننده میگذرد و بین استک و رجیستر استک وجود دارد، در نظر گرفتیم بنابراین زمان مسیر بحرانی برابر است با:

$$\text{critical_time(Register)} + \text{critical_time(Subtractor)} = 2.5 + 10 = 12.5$$

رجیستر 3 بیتی شامل 6 گیت، 3 مالتی پلکسر 4 به 1 و 3 DFF می باشد، که مسیر بحرانی آن شامل گیت ها(0.5) و یک مالتی پلکسر 4 به 1(2) می باشد(زیرا رجیستر های 1 بیتی در رجیستر 3 بیتی cascade شده به صورت موازی کار میکنند و تاخیر آنها نباید با هم جمع شود) پس در کل تاخیر مسیر بحرانی این رجیستر $0.5+2 = 2.5$ می باشد.

طبق مدار cascade شده نشان داده شده برای subtractor، برای محاسبه تاخیر این ماجول باید تاخیر full adder و not را در نظر بگیریم. برای تاخیر full adder هم باید تاخیر 3 ماجول carry و sum و not را در نظر بگیریم. ماجول not با توجه به اینکه با سلول C1 ساخته میشود تاخیر 2 و ماجول Carry چون با سلول C1 ساخته میشود تاخیر 2 و sum تاخیر $2+2=4$ دارد. بنابراین با در نظر گرفتن مدار cascade شده تاخیر کلی subtractor برابر 10 خواهد بود. پس مجموع تاخیر این دو ماجول یعنی 12.5 مقدار نهایی است.

مساحت کل:

- Comparator: $3 * (2 * \text{area}(C1) + \text{area}(C2)) = 3 * (2 * 7 + 8) = 66$
- Register: $3 * \text{area}(S2) = 3 * 15 = 45$
- Multiplexer: $3 * \text{area}(C1) = 3 * 7 = 21$
- Accumulator: $5 * (S2) + 5 * (C1) = 3 * 15 + 5 * 7 = 80$
- Subtractor: $3 * (3 * \text{area}(C1) + (\text{area}(C2))) = 3 * (3 * 7 + 8) = 87$

$$299 = 87 + 80 + 21 + 45 + 66 \text{ مسیر داده:}$$

- S2+: $\text{area}(S2) + 3 * \text{area}(C1) = 15 + 3 * 7 = 36$
- S1+: $\text{area}(S2) = 15$
- S0+: $\text{area}(S2) + 3 * \text{area}(C1) = 15 + 3 * 7 = 36$
- Sel_reg: $2 * \text{area}(C1) = 14$
- Ld: $\text{area}(C2) + \text{area}(C1) = 8 + 7 = 15$
- Sel_cmp: $\text{area}(C1) = 7$
- Sel_sub: $\text{area}(C1) = 7$
- Push: $\text{area}(C1) = 7$
- Pop: $\text{area}(C1) = 7$
- En: $\text{area}(C1) = 7$
- Done: $2 * \text{area}(C1) = 14$

$$165 = 7 * 5 + 14 + 15 + 14 + 36 + 36 + 15 \text{ مساحت کلی کنترلر:}$$

$$\text{مساحت کلی مدار: } 464 = 299 + 165$$

زمان اجرا:

$$\text{Running_time} = \text{critical_time} * \text{number_of_clocks} = 12.5 * 184 = 2300$$

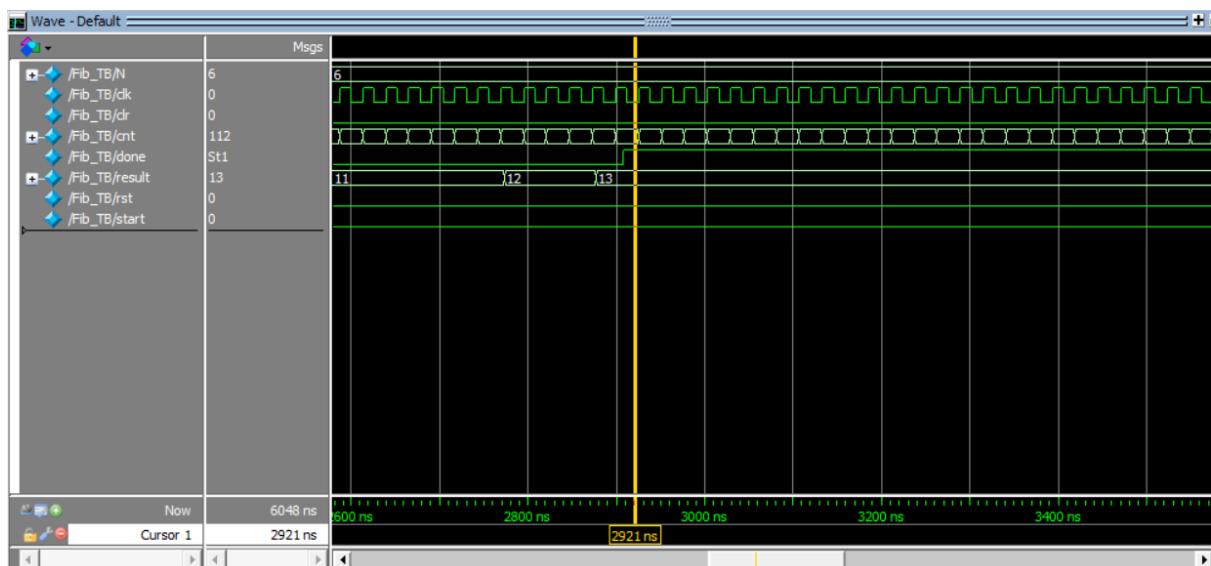
مقدار number_of_clocks از روی waveform محاسبه شد. که برای این کار یک counter قرار دادیم و هنگامی که سیگنال done یک شده، مقدار شمارنده برابر 184 بود:



همچنین همانطور که دیده می شود مقدار خروجی به ازای $N = 7$ برابر 21 شده و درست است.

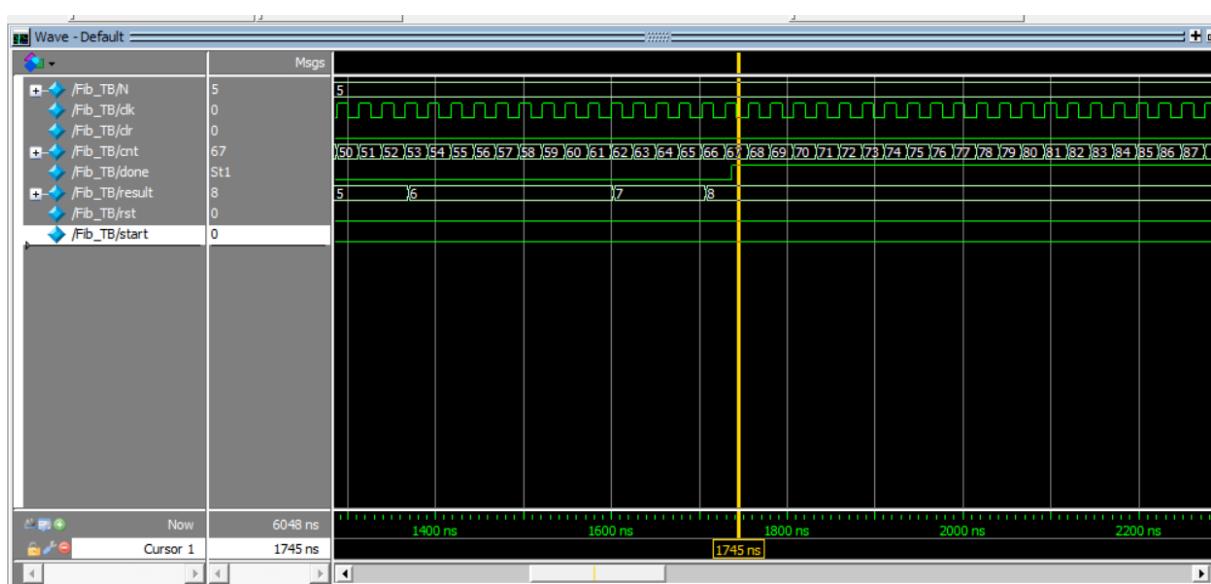
چند تست دیگر:

N = 6:



مقدار خروجی برابر 13 شده و درست است.

N = 5:

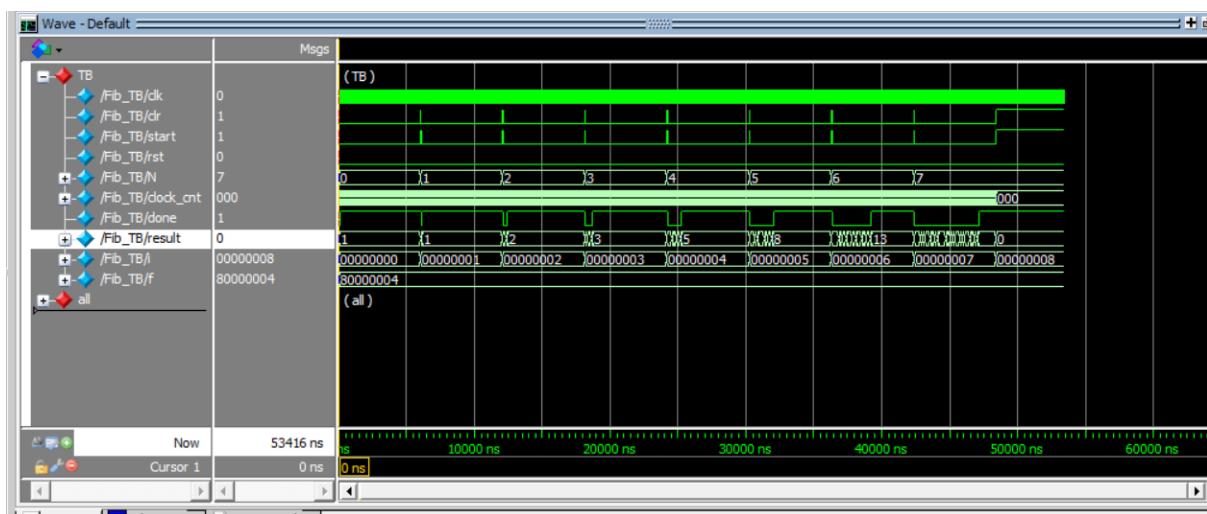


مقدار خروجی برابر 8 شده و درست است.

Testbench

برای پیاده سازی تست بنچ گفته شده در پروژه، ابتدا یک آرایه به نام `expected` تعریف می کنیم که در آن خروجی های مورد انتظار تابع `fibonacci` را به ازای تمامی مقادیر ورودی ۰ تا ۷ با استفاده از یک بلاک `initial`، ذخیره می کنیم. سپس با استفاده از تابع `fopen` در یک بلاک `initial` یک فایل جدید به نام `result.txt` را باز می کنیم. در نهایت روی اعداد ۰ تا ۷ با استفاده از حلقه `for` پیمایش می کنیم و هربار خروجی مورد نظر را با استفاده از تابع `fwrite` در فایل می نویسیم.

در نهایت خروجی کل را میتوان به صورت زیر دید:



* توجه شود که در مرحله پیاده سازی در کد وریلاغ با توجه به اینکه time scale کد را به صورت $1\text{ns}/1\text{ns}$ گذاشته بودیم، دقت مان اعشاری نمی باشد. پس ما جدول تاخیر اجزای مدار را که در صورت پروژه داده شده بود در 2 ضرب کردیم تا همه تاخیرها به صورت عدد صحیح

باشند. به همین خاطر طول کلاک را در کد وریلاغ به جای 12.5 مقدار 25 در نظر گرفتیم.

در این صورت زمان اجرایی که در بالا به صورت تئوری محاسبه کردیم به جای 2300 مقدار

4600 خواهد بود.