

درس معماری کامپیوتر

تمرین سری دوم

۱۴۰۳/۱۲/۲۵

متن باقری

۴۰۲۱۰۵۷۲۷

(۱) point to point: اگر فرض کنیم هر ثابت ۱ بیتی باشد، به 32 MUX و 32 سیم نیاز خواهیم داشت. اکنون که هر ثابت ۳۲ بیت دارد، به $32 \times 32 = 1024$ سیم تک‌بیتی و $32 \times 32 = 1024$ MUX نیاز داریم. همچنین به $5 \times 32 = 160$ سیگنال کنترلی برای MUX ها نیاز داریم.

Common bus: در این حالت اگر ثابت ها دارای سیگنال load enable و output enable باشند 2، به یک 32 bus بیتی و $64 = 32 \times 2$ سیگنال enable نیاز خواهیم داشت. اگر ثابت ها سیگنال output enable نداشته باشند 3، به ۳۲ MUX و 5 سیگنال کنترلی برای آن ها و 32 سیگنال کنترلی برای ثابت ها نیاز داریم.

(ب) با توجه به توضیحات داده شده و شماره‌گذاری که برای حالت های بخش الف انجام دادیم:

1: $64 \times 32 = 2048$ سیم تک بیتی، $64 \times 32 = 2048$ MUX، $6 \times 64 = 384$ سیگنال کنترلی برای MUX ها

2: یک 64 bus بیتی و $64 \times 2 = 128$ سیگنال enable

3: ۳۲ MUX و 6 سیگنال کنترلی برای آن ها و 64 سیگنال کنترلی برای ثابت ها

(ج) با فرض اینکه تعداد ثابت ها همان 32 است:

1: $64 \times 32 = 2048$ سیم تک بیتی، $64 \times 32 = 2048$ MUX، $5 \times 32 = 160$ سیگنال کنترلی برای MUX ها

2: یک 32 bus بیتی و $32 \times 2 = 64$ سیگنال enable

assuming reset is 0. (۲) آ

(B): $R1 \leftarrow A$

(C): $E \leftarrow R2 + 2$

(!C): $E \leftarrow R1$

$R2 \leftarrow E$

registers initial values are supposed to be 0. (ب)

<i>Time (ns)</i>	<i>R1</i>	<i>E</i>	<i>R2</i>
0	0	2	0
10	5	4	2
20	19	19	4
30	19	21	19
40	11	11	21
50	11	13	11

(۳) آ

A: $R0 \leftarrow R0 + 1$

B: $R2 \leftarrow R0 \text{ xor } R2, \quad R0 \leftarrow R0 + 1$

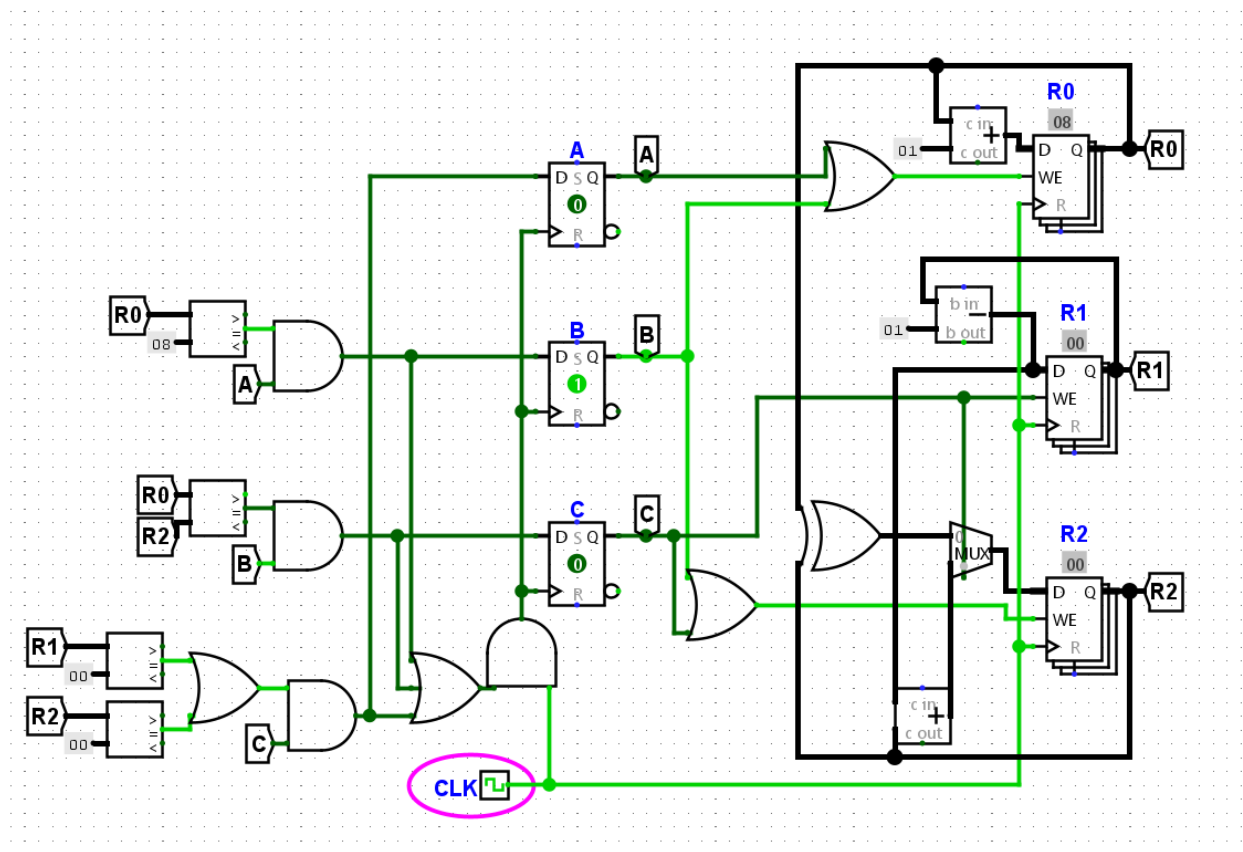
C: $R1 \leftarrow R1 - 1, \quad R2 \leftarrow R2 + R1 - 1$

A & (R0 = 8): $A \leftarrow 0, \quad B \leftarrow 1, \quad C \leftarrow 0$

B & (R0 = R2): $A \leftarrow 0, \quad B \leftarrow 0, \quad C \leftarrow 1$

C & (R1 = 0 | R2 = 0): $A \leftarrow 1, \quad B \leftarrow 0, \quad C \leftarrow 0$

ب) مدار طراحی شده به این شکل است:



فایل این مدار با نام T_Q3 در همین پوشه موجود است.

۴) آ دستورات را از بالا به پایین و از 0 تا 5 شماره گذاری کردیم.

<i>cycle</i>	<i>Instruction no.</i>	<i>R0</i>	<i>R1</i>	<i>R2</i>
0 (initial state)	—	0	5	1
1	0	1	5	6
2	1	2	4	6
3	2	3	4	12
4	3	4	4	0
5	4	0	4	0

6	0	1	4	4
7	1	2	3	4
8	2	3	3	8
9	3	4	3	2
10	4	0	3	2
11	0	1	3	5
12	1	2	2	5
13	2	3	2	10
14	3	4	2	0
15	4	0	2	0
16	0	1	2	2
17	1	2	1	2
18	2	3	1	4
19	3	4	1	0
20	4	0	1	0
21	0	1	1	1
22	1	2	0	1
23	5	_____	terminated	_____

$R0 = 2$ $R1 = 0$ $R2 = 1$ برنامه پس از ۶ کلاک متوقف شده و مقادیر نهایی:

ب) 23 سیکل زمانی

ج) در ستون instruction no. شماره دستور اجرا شده مشخص شده است. دستورات را از بالا به پایین و از 0 تا 5 شماره گذاری کردیم.

د) با در نظر نگرفتن حالتی که مقدار اولیه ثباتها در آن نوشته شده (سایکل 0):

1 مرتبه حالتی که مقدار اولیه ثبات‌ها در آن نوشته شده (سایکل 0) + 4 مرتبه $R0 = 0$,

5 مرتبه $R0 = 1$,

5 مرتبه $R0 = 2$,

4 مرتبه $R0 = 3$,

4 مرتبه $R0 = 4$,

(۵) آ) الگوریتم اقلیدسی، با این تفاوت که به جای تقسیم از تفریق‌های متوالی استفاده می‌کنیم:

$$(R1 > R2) : R1 \leftarrow R1 - R2$$

$$(R2 > R1) : R2 \leftarrow R2 - R1$$

$$(R1 = R2) : R3 \leftarrow R1$$

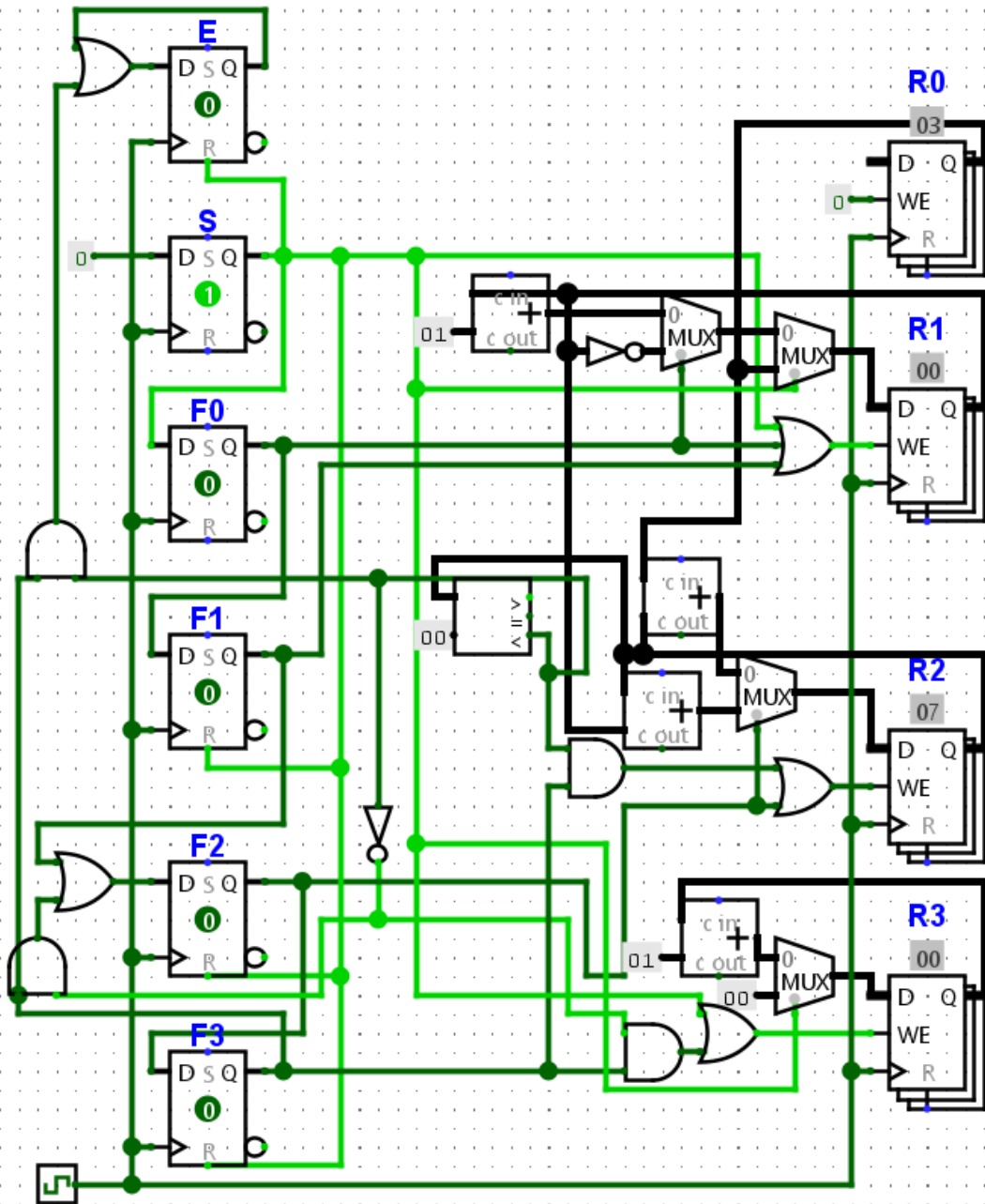
(۶) آ) با صفر نبودن S خط اول اجرا شده، S صفر می‌شود و F0 یک می‌شود تا در مرحله بعد خط دوم اجرا شود. کلاً در هر مرحله شرط این مرحله صفر شده و شرط مرحله بعد یک می‌شود تا خطوط به ترتیب اجرا شوند. $R0$ نقیض شده، با یک جمع شده (یعنی مکمل دو می‌شود) و در $R1$ ریخته می‌شود. سپس $R2$ با $R1$ جمع می‌شود. $R2 \leftarrow R2 - R0$. اگر حاصل منفی بود، $R2$ به مقدار اولیه خود در زمان شروع برنامه بر می‌گردد و برنامه تمام می‌شود. در غیر این صورت $R3$ یکی زیاد شده و به خط F2 بر می‌گردیم. این برنامه در واقع خارج قسمت تقسیم $R2$ بر $R0$ را در $R3$ و باقی‌مانده تقسیم را در $R2$ ذخیره می‌کند.

$n = 4$ $R0 = 2$ $R2 = 7$

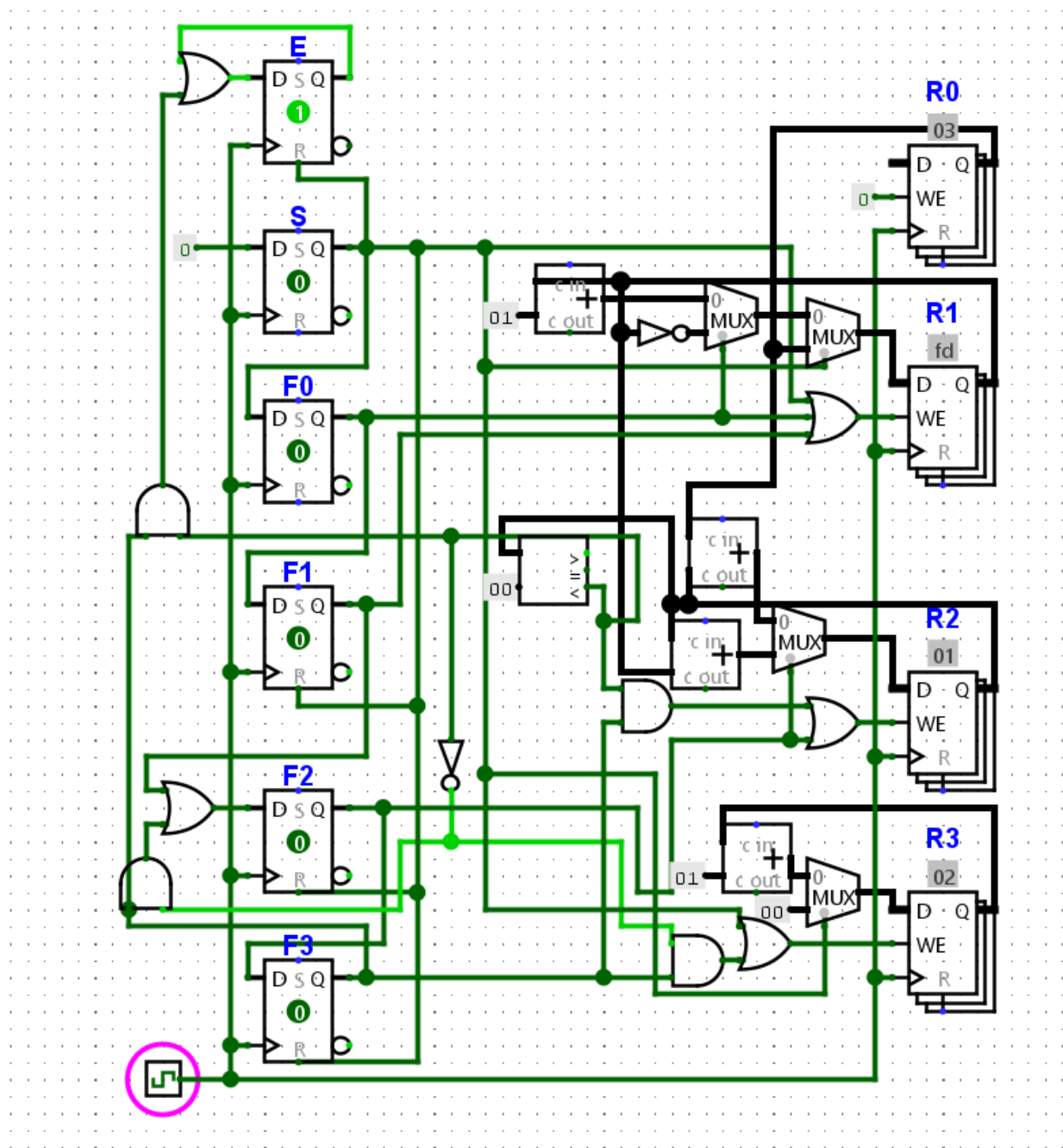
مثال ساده:

<i>state</i>	<i>R1</i>	<i>R2</i>	<i>R3</i>
<i>S</i>	2 (= 0010)	7	0
<i>F0</i>	(1101)	7	0
<i>F1</i>	-2 (= 1110)	7 (0111)	0
<i>F2</i>	-2	5 (= 0101)	0
<i>F3</i>	-2	5	1
<i>F2</i>	-2	3	1
<i>F3</i>	-2	3	2
<i>F2</i>	-2	1	2
<i>F3</i>	-2	1	3
<i>F2</i>	-2	-1	3
<i>F3</i>	-2	1	3

مدار طراحی شده در حالت اولیه به این شکل است، در این مثال ۷ را بر ۳ تقسیم کردیم.

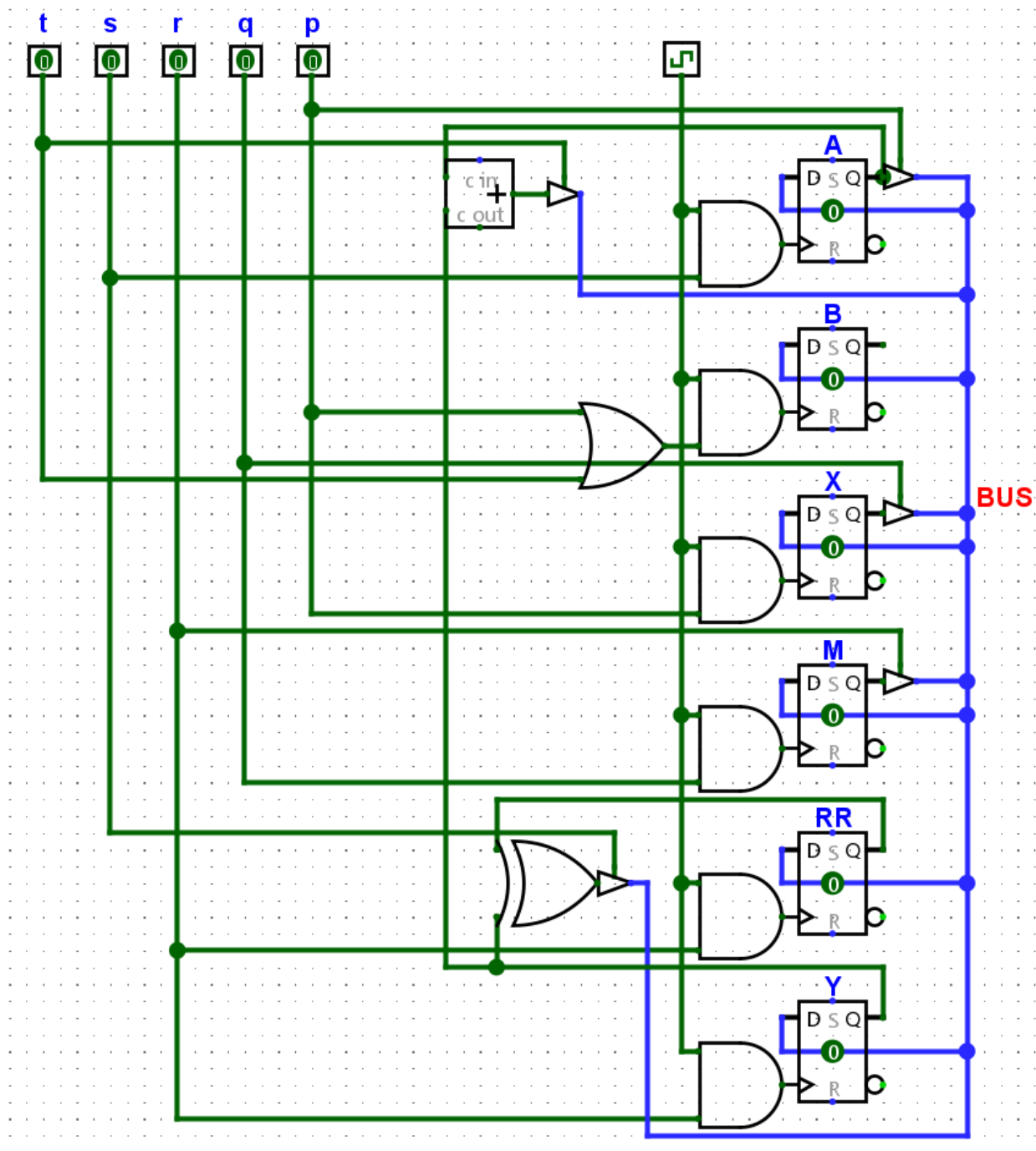


مدار در حالت نهایی به این شکل است. همانطور که مشاهده می‌شود خارج قسمت ۲ و باقی‌مانده ۱ است.



فایل این مدار با نام T_Q6 در همین پوشه موجود است.

۷) مدار طراحی شده به این شکل است:



در این تصویر رنگ آبی نشان دهنده گذرگاه مشترک است که ورودی همه فلیپ‌فلاپ‌ها به آن متصل اند. همچنین برای کنترل نوشتن بر روی گذرگاه از buffer سه حالت استفاده شده. به دلیل عدم وجود ورودی سیگنال Write Enable بر روی فلیپ‌فلاپ‌ها، از گیت های AND برای کنترل کلاک آن‌ها استفاده شده.

فایل این مدار با نام T_Q7 در همین پوشه موجود است.

=====