

از دلیلهایی که در CA قبل برای لیت مختلف استفاده کردیم. برای قیمت های ① تا ⑤
 مدار TCS، یک block always نوشتیم و دلیلهایی که در آن قیمت بود را در block always گنجاندیم
 (لازم توجه است که چون در پرده قبلی از delay to delay گیت ها استفاده کردیم، در این CA هم همان کار را کردیم)

① → 19 NS : هر XOR، 7 NS و 12 NS دلیله دارد : 19 NS
 ② 25 NS : 3 input NOR و NOT دلیله 25 NS
 ③ 35 NS : با درن محاسباتی بودن احتمال تغییر کردن در لیت، بیت های a_1 و b_1 با احتمال $\frac{1}{4}$ تغییر کرده و در لیت قیمت ③، تاخیر حاصله از آن ها، برابر جمع یک XOR، NOT، و 3 input NAND است

$$\frac{2}{4}(19+7+12+8) = 23$$

$$\frac{1}{4}(8+7+12) \approx 7 NS$$

$$\frac{1}{4}(8+12) = 5$$

$$\Rightarrow \text{total delay} = 23 + 5 + 7 = 35 NS$$

④ 8 NS : دلیله یک 2 input NAND تاخیر ندارد است. ⑤ دلیله یک NOT، 2 input NAND، 12 NS

② 25 NS : 3 input NOR و NOT ریلی ، هر XOR ، 2 input NAND و 17 NS
 ③ 35 NS : با فرض مساوی بودن احتمال تغییر کردن هر بیت ، بیت های a_1 و b_1 با احتمال $\frac{1}{4}$ تغییر کرده و در هر وقت ③ ،
 تاخیر حاصله از آن ها ، برابر جمع یک XOR ، NOT ، 3 input NAND و 2 input NAND است \leftarrow
 $\frac{2}{4}(19+7+12+8) = 23$ NS
 همچنین بیت a_0 با احتمال $\frac{1}{4}$ تغییر کرده و ریلی حاصله از آن : $\frac{1}{4}(8+7+12) = 7$ NS
 $\frac{1}{4}(8+12) = 5$ NS
 $\Rightarrow \text{total delay} = 23 + 5 + 7 = 35 \text{ NS}$

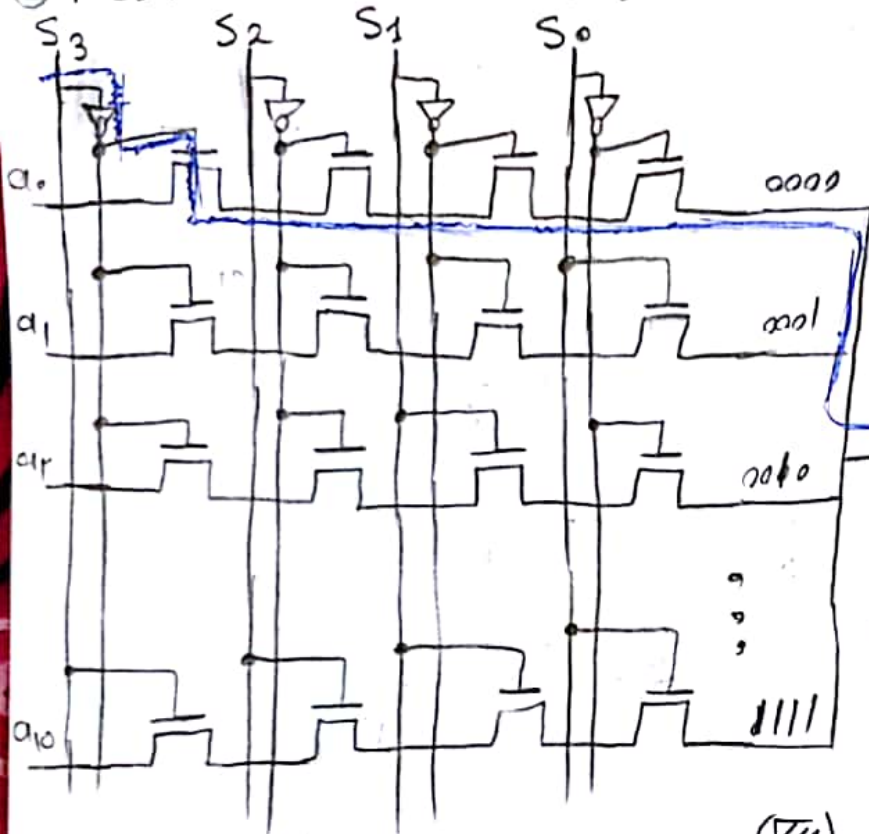
④ 8 NS : ریلی یک 2 input NAND تاخیر ندارد است . ⑤ \leftarrow ریلی یک NOT و 2 input NAND : 12 NS

Q3 : با اجرای تست بنچ و اجرای test vectors مختلف مقادیر ورودی به دست آمده :

	to 0	to 1	
EQ	119	119	NS
GT	91	91	NS

Q4-A : برای پیاده سازی TMA و TMI به صورت MUX-based logic ، ما از یک CMOS-4to1-MUX و یک CMOS-16to1-MUX و pass transistor-16to1-MUX استفاده کردیم (من بعد)
 Q7-A :

1) pass transistor - 16 to 1 - MUX :



حال delay هر کدام را حسابی کنیم :

path A در میان path های موجود دارای بیشترین transistor در سیر راه خودش است

و delay آن برابر Path A است با 5 NOT و 4 AND nmos transistors

$$\text{delay} : 7 + 4 \times 5 = 27 \text{ ns}$$

to NOT delay nmos to Z delay

II) CMOS - 16 to 1 - MUX and CMOS - 4 to 1 - MUX :

design ما برای ساختن این MUX به صورت 2 level logic AND-OR است که تبدیل به تمام NAND می شود . در level اول تعدادی 5 input NAND قرار دارند و در level دوم ، خروجی این NAND ها به 4 تا select input و یکی از 16 ورودی

ها که 14 تا هستند ، ورودی یک 16 input NAND قرار می گیرند . اگر مدار 1 و 2 را به صورت complex

$$① \quad 5 \times 4 = 20 \text{ ns}$$

to NOT delay nmos

بازیم ، delay گیت ها به صورت دوم رو خواسته بودیم

(H) CMOS-16to1-MUX and CMOS-4to1-MUX: (A)
 design ما برای ساختن این MUX به صورت 2 level logic AND-OR است که تبدیل به تمام NAND gate می شود. در level اول تعدادی 5 input NAND قرار دارند و در level دوم، خروجی این NAND ها که 4 تا select input و یکی از 16 ورودی (1)

ها که 14 تا هستند، ورودی یک 16 input NAND قرار می گیرند. اگر مدار (1)، را به صورت complex (2)

بازیم، delay که به صورت دو رو خواسته بود، از آنجا که complement یعنی select ها، ورودی level اول هستند پس delay که NOT هم اضافه می شود

① $5 \times 4 = 20 \text{ NS}$
 num of nmos to delay nmos

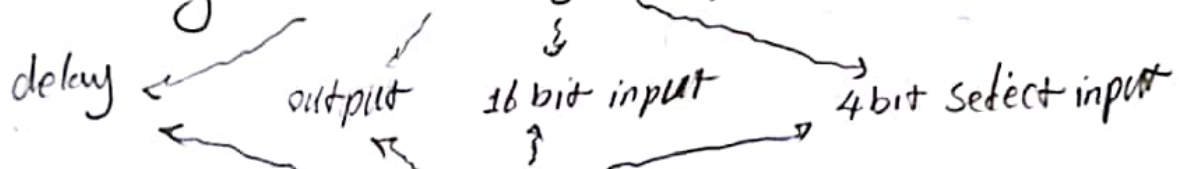
② $16 \times 4 = 64 \text{ NS}$

total delay: $7 + 20 + 64 = 91 \text{ NS}$

(A) همانند طراحی مدار (A): level 1: $3 \times 4 = 12 \text{ NS} \leftarrow 3 \text{ input NAND}$
 $7 \text{ NS} \leftarrow \text{to NOT}$
 level 2: $4 \times 4 = 16 \text{ NS} \leftarrow 4 \text{ input NAND}$
 total delay: $7 + 12 + 16 = 35 \text{ NS}$

حال برای نوشتن assign statement ها از نوع behavioral نوشتن استفاده می کنیم:

PT-16to1-MUX: assign #27 $\bar{w} = J[S]$;



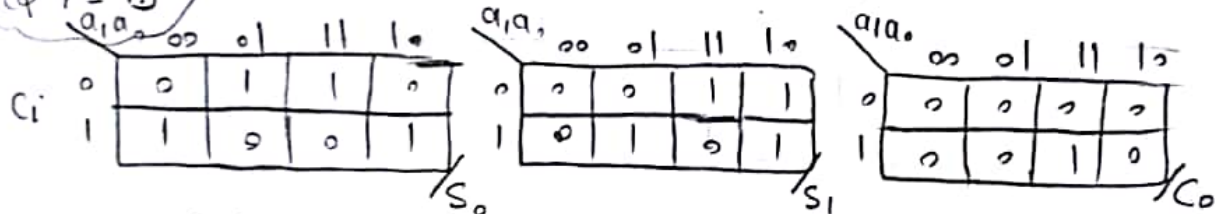
CMOS-16to1-MUX: assign #20 $\bar{w} = J[S]$;

CMOS-4to1-MUX: assign #35 $\bar{w} = J[S]$; ($J \rightarrow$ 4 bit input
 $S \rightarrow$ 2 bit select input)

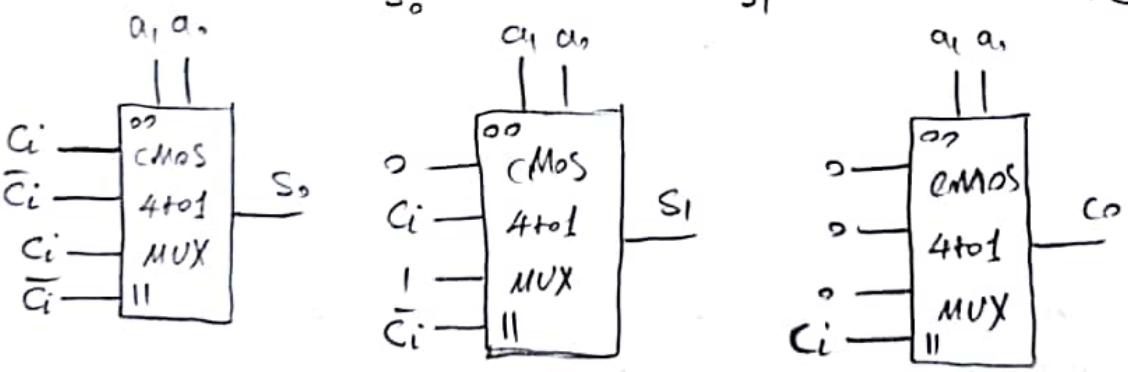
Q 4-B

Q 7-B

: جزئیات طراحی TMA با استفاده از MUX ها، در استفاده تلفیق شماره وجود است



طراحی TMI :



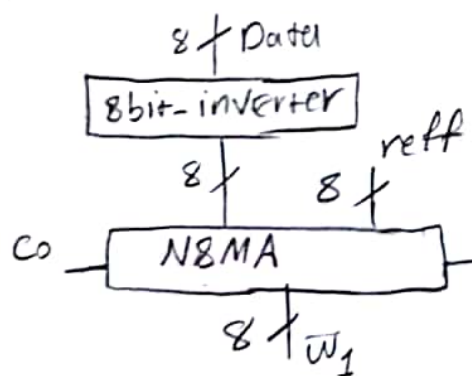
Q 6:

Q 9:

باتوجه به مقدارهای بدست آمده برای (لیستی) N8MA و N8MI در تست پنج هاداریم

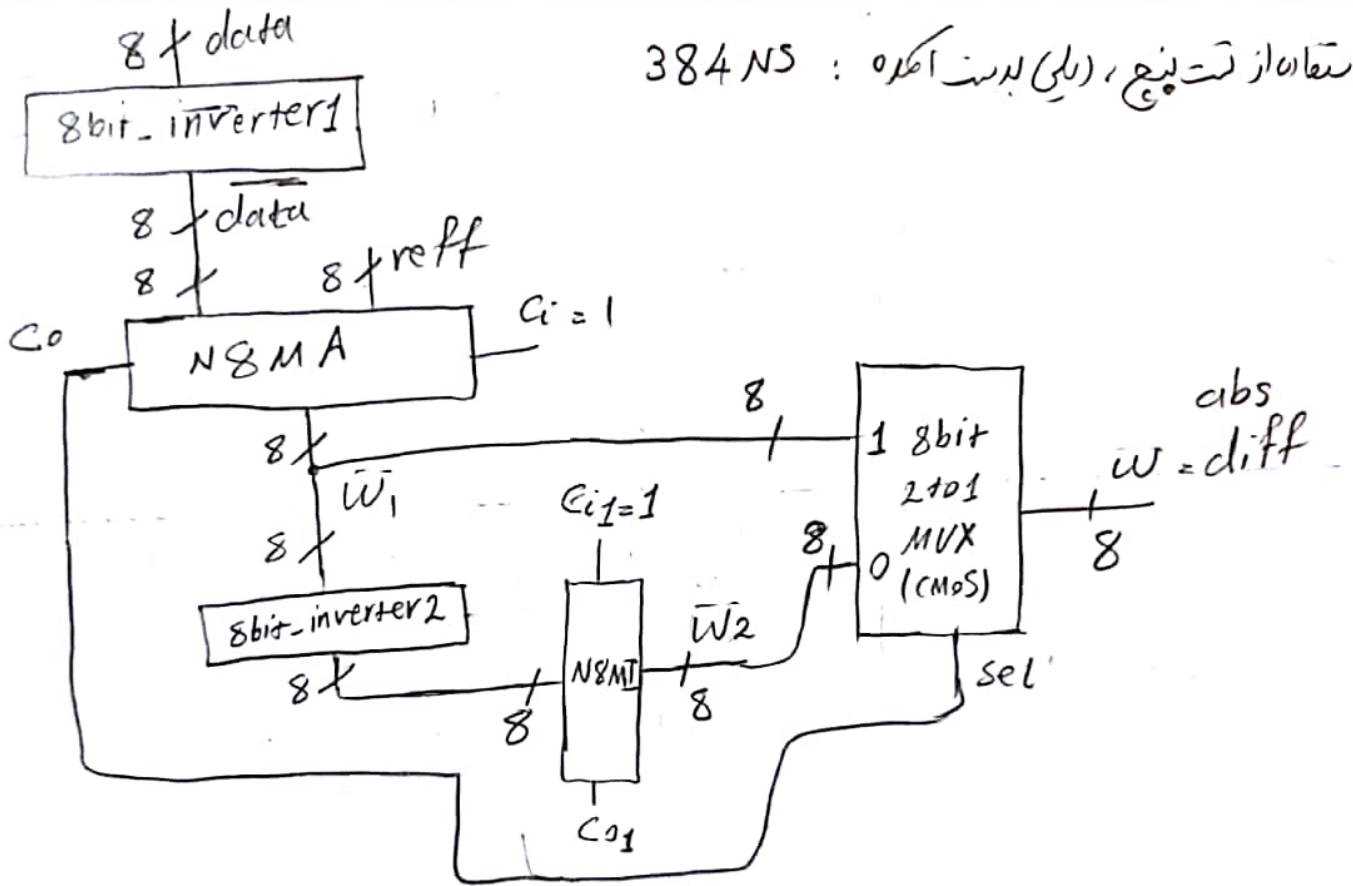
worst case delay N8MA : 360NS worst case delay N8MI : 147NS

طراحی Abs diff : واضح است که نیاز به عمل subtract داریم. برای این کار ابتدا باید 2^{com} (generate statement) یکی از input ها را بهت بیاوریم. پس ابتدا یک 8bit-inverter طراحی کردیم (بالای not) و سپس خروجی آن را با ورودی اول داخل مدار N8MA بردیم با $ci = 1$ اینگونه 2^{com} ورودی دوم را با

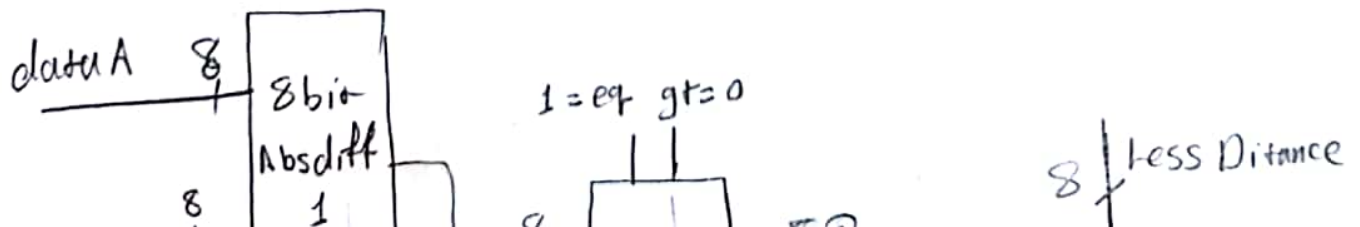


ورودی اول جمع زده ایم یعنی عمل subtract : $\{ w = reff - Data \}$
 حال از یک 8bit-2to1-mux استفاده می کنیم و اگر co ، 1 شد (یعنی) با بیت 9ام فرضی (LMB) 2^{com} ورودی data که 1 جمع شده و حاصل w مثبت است) و اگر ورودی خروجی می بردیم وگرنه 2^{com} آن را (کل مدار صفحه بعد)

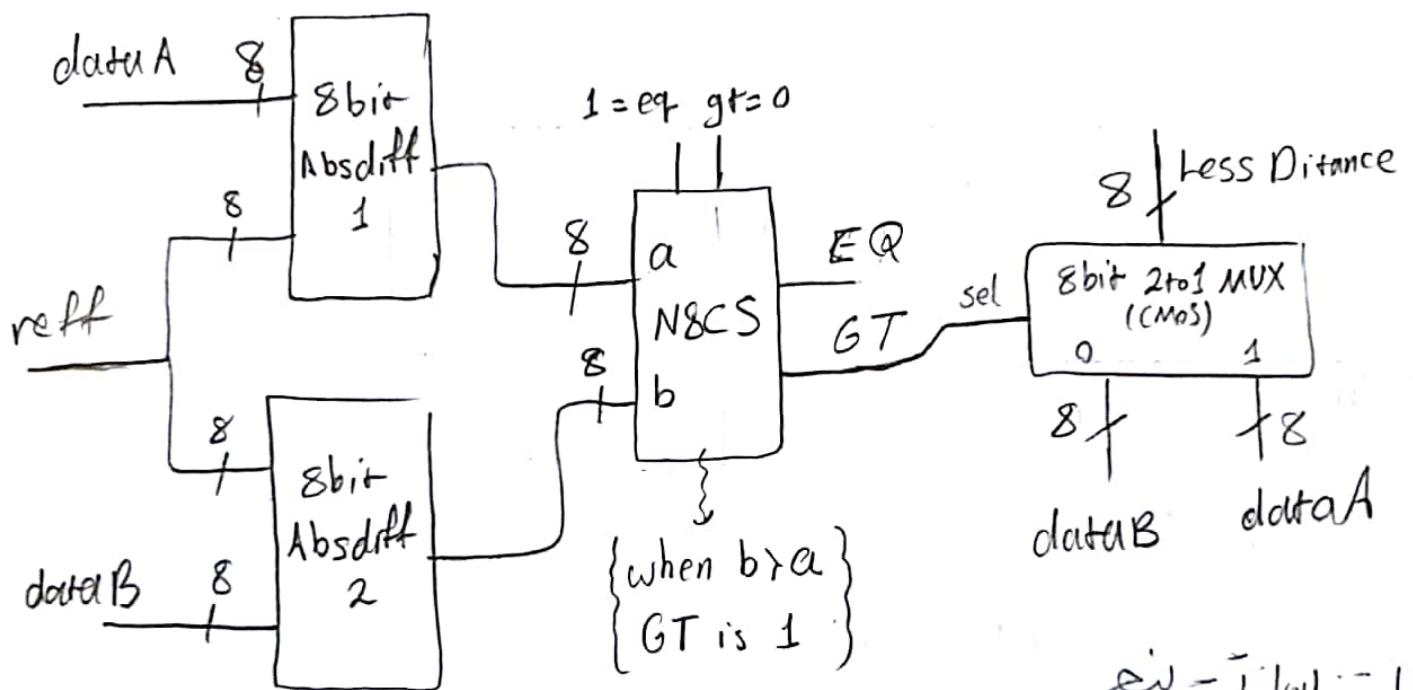
با استفاده از تست پنج، (دلی بدست آمده) : 384 NS



Q 12, 13 :



Q 12, 13 :



با استفاده از تست بیج

521ns : دلی

assign statements of Q 14 :

```

assign distA = (reff > dataA) ? reff - dataA : dataA - reff ;
assign distB = (reff > dataB) ? reff - dataB : dataB - reff ;
assign #520 W = (distB > distA) ? dataA : dataB ;

```