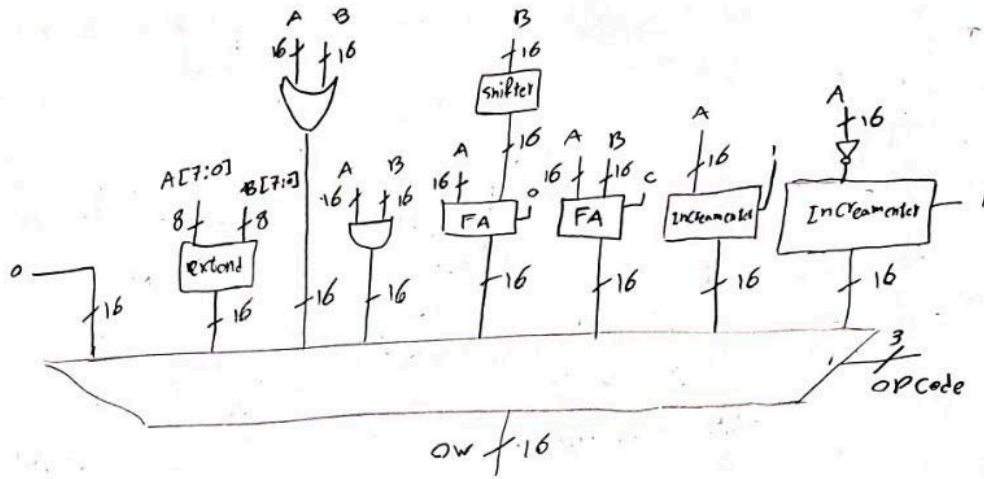


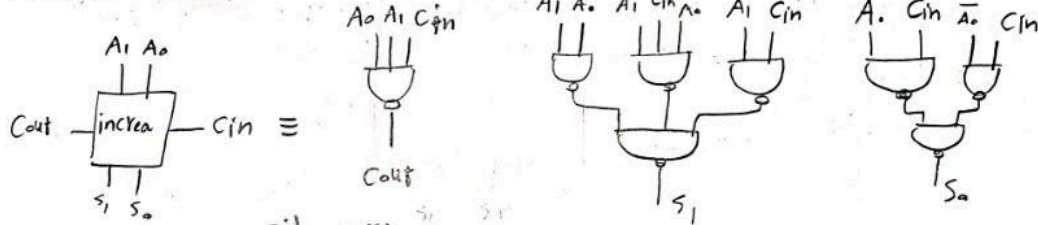
سوال 1) در ابتدا شکل مدار را نشان میدهیم:

①



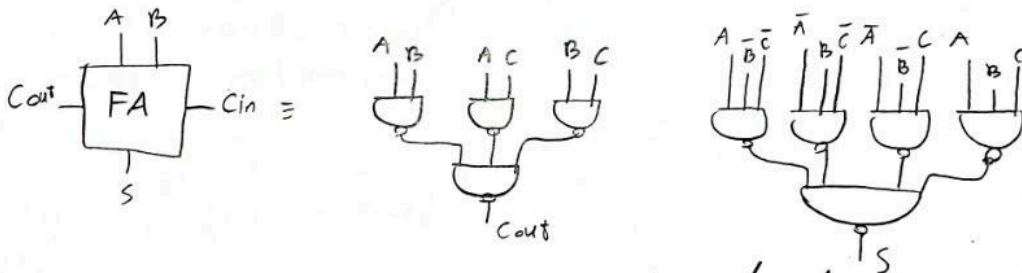
برای هر یک از کسکده های زیر که قابل کسکده است:

A) Incrementer



کسکده آن هم به صورت زیر می باشد: Cout به Cin یونیت می آید

B) Full Adder



کسکده آن مانند کسکده زیر می باشد.

برای شیفتر هم نامی است به نام ویرول را به بیس ۱- نامی از طریق به بیس ۸ و ۱۶

d) extend:

مانند شیفتر عمل می کنند فقط به ۷ بیتی (A[7:0]) و به ۱۵ بیتی (A[7:0]) و به ۱۵ بیتی (A[7:0])

قسمت (a) کد ها نوشته شده و قابل مشاهده است. ویو فرم ها در بخش c.

قسمت (b) به عکس زیر توجه کنید:

```
=== ALU_behave ===
```

Number of wires:	1122
Number of wire bits:	1169
Number of public wires:	7
Number of public wire bits:	54
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	633
NAND	262
NOR	285
NOT	86

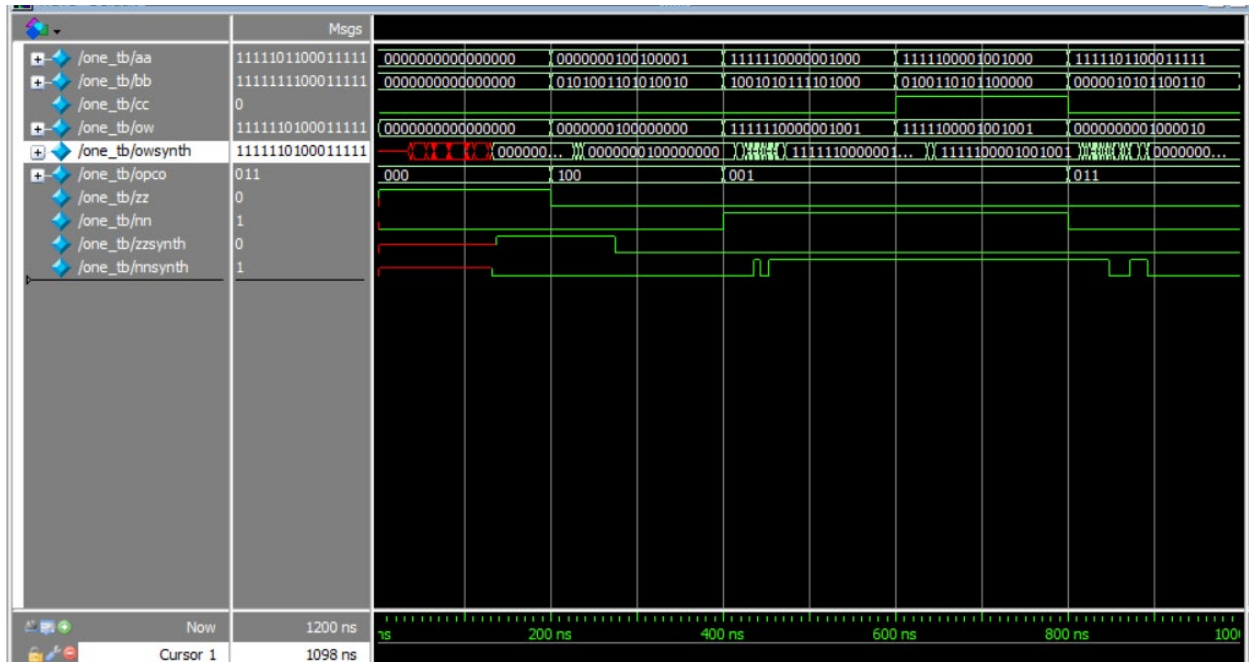
که میتوان به فایل سنتز شده رفت و گیت ها را مشاهده کرد:

و همانطور که مشاهده میکنید از 633 گیت در این ساختار سنتز

شده استفاده شده است.

```
NOT _1089_ (  
  .A(_1023_),  
  .Y(_1024_)  
);  
NOR _1090_ (  
  .A(_0452_),  
  .B(_0436_),  
  .Y(_1025_)  
);  
NOR _1091_ (  
  .A(_1024_),  
  .B(_1025_),  
  .Y(_1026_)  
);  
NAND _1092_ (  
  .A(_0468_),  
  .B(_1026_),  
  .Y(_1027_)  
);  
NOR _1093_ (  
  .A(_0468_),  
  .B(_1026_),  
  .Y(_1028_)  
);
```

قسمت c) در این قسمت، ویو فرم های متعلق به پس از عملیات سنتز با پسوند synth جدا شده اند (هم ویو فرم های حالت معمولی هست هم حالت سنتز شده توسط yosys):

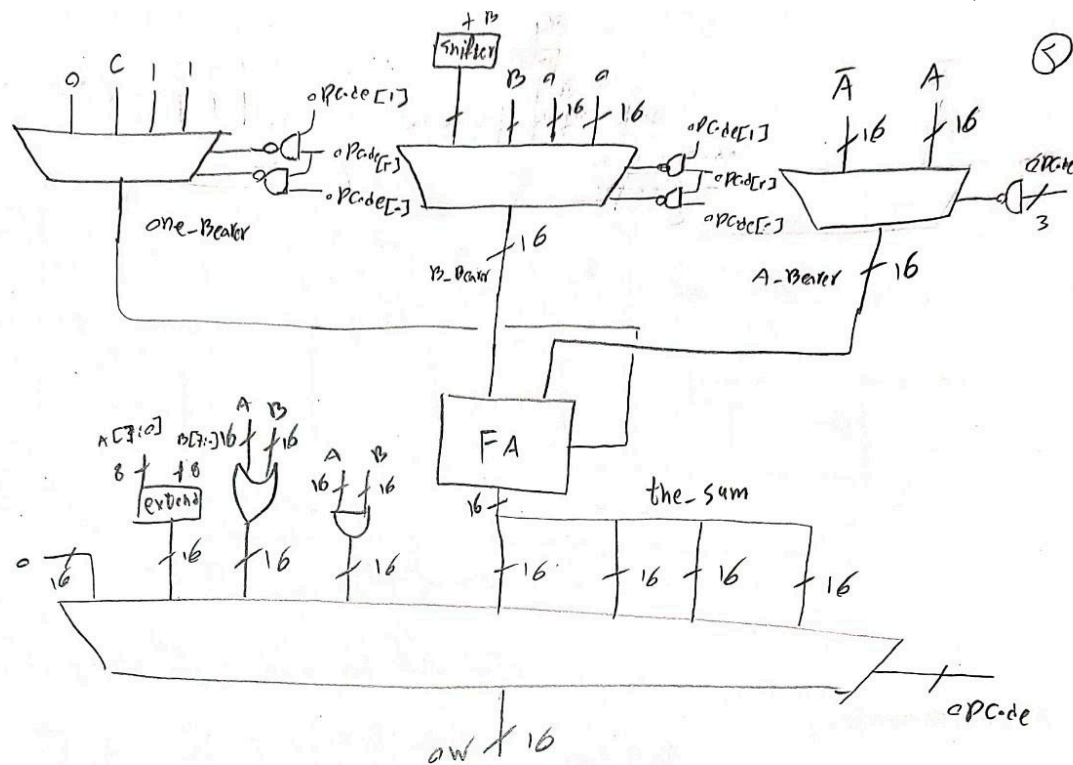


قسمت d) همانگونه که در شکل موج های بالا مشخص می باشد، حالت معمولی دلیلی نداشته و بلافاصله شکل موج ها عوض میشوند. اما ویو فرم سنتز شده توسط یوسیس دارای دلیلی نسبتاً بالایی است، که این امر به دلیل نحوه کسکیدینگ و ساختار چند لایه مدار سنتز شده به دلیل نبود گیت های با بالای دو ورودی می باشد که یوسیس برای حل این مشکل، از شکستن گیت های چند ورودی و باسازی گیت ها به کمک گیت نات و نند یا نور دو اینپوت استفاده میکند و همین امر سبب بالا رفتن ورست کیس دلیلی میشود. همان گونه که مشاهده میشود، جواب نهایی هر دو مدار درست و با یکدیگر برابر است. در خروجی های یک بیتی ساختار سنتز شده گاهی هازارد مشاهده میشود.

سوال 2)

قسمت a) در این سوال سعی میکنیم با دست بردن در مدارمان و باز استفاده کردن ساختار های تکراری، از تعداد گیت ها بکاهیم. در ابتدا به شکل این مدار

توجه کنید:



همانگونه که مشاهده میکنید با جا به جایی ترتیب مدار و چند عملیات پیش پردازش روی ورودی های مدار، توانستیم چهار ورودی ابتدایی alu را با تعداد rti کمتری باز سازی کنیم.

قسمت (b) برای مقایسه با مدار ساده سوال یک، به گزارش یوسیس توجه کنید:

```

=== ALU_struct_2 ===

Number of wires:      874
Number of wire bits:  921
Number of public wires: 7
Number of public wire bits: 54
Number of memories:   0
Number of memory bits: 0
Number of processes:  0
Number of cells:      449
    NAND              230
    NOR                154
    NOT                65
    
```

تعداد گیت ها به 449 تا کاهش یافت، که مقدار قابل توجهی است! از طرفی سری به کد وریلاگ ساخته شده هم بزنیم:

```
NOR _0987_ (
  .A(_0067_),
  .B(_0112_),
  .Y(_0113_)
);
NAND _0988_ (
  .A(_0111_),
  .B(_0113_),
  .Y(_0114_)
);
NAND _0989_ (
  .A(_0066_),
  .B(_0114_),
  .Y(_0115_)
);
NAND _0990_ (
  .A(_0059_),
  .B(_0115_),
  .Y(_0116_)
);
NOT _0991_ (
  .A(_0116_),
  .Y(_0117_)
);
```

اما پیش از رفتن به قسمت بعد ، لازم میدانم نتایج دو آزمایش دیگر را هم ارائه کنم. اولین آزمایش با تغییر دادن شیفترا ها و فول ادر ها و استفاده ابتدایی هر کدام انجام شد (در کد 1-2 قابل مشاهده است) که نتایج مطلوبی را در بر نداشت. به گزارش یوسیسی توجه کنید:

```
=== ALU_struct_1 ===

Number of wires:          1225
Number of wire bits:      1287
Number of public wires:   8
Number of public wire bits: 70
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          703
    NAND                  278
    NOR                    324
    NOT                    101
```

همانطور که مشاهده میکنید، تعداد گیت ها از حالت پایه بیشتر هم شد! دومین رویکرد همین رویکردی است که پیشتر در مورد آن صحبت کردیم که کد آن در فایل 2-2.sv قابل مشاهده است. سومین رویکرد در واقع به این صورت است که از یک باس 33 بیتی استفاده میکنیم، به طوری که 16 بیت اول آن نگه دارنده مقدار a یا نقیض a و مقادیر سه آپکود آخر، 16 بیت دوم آن نگه دارنده b، شیفت b و صفر و بیت آخر نگه دارنده صفر، یک یا c خواهد بود. این باس را به یک مولتی پلکسر هشت به یک میدهیم و مقدار هر بخش آن را با توجه به هر آپکود مشخص میکنیم، در نهایت هر سه بخش آن را به یک فول ادر میدهیم و مقدار جمع شده را روی خروجی میگذاریم. در این روش تعداد قطعات خیلی کمتری مصرف میشود، ولی از آنجایی که یک باس 33 بیتی داریم، مولتی پلکسر بزرگی خواهیم داشت که گیت های زیادی لازم دارد. به گزارش یوسیس توجه کنید:

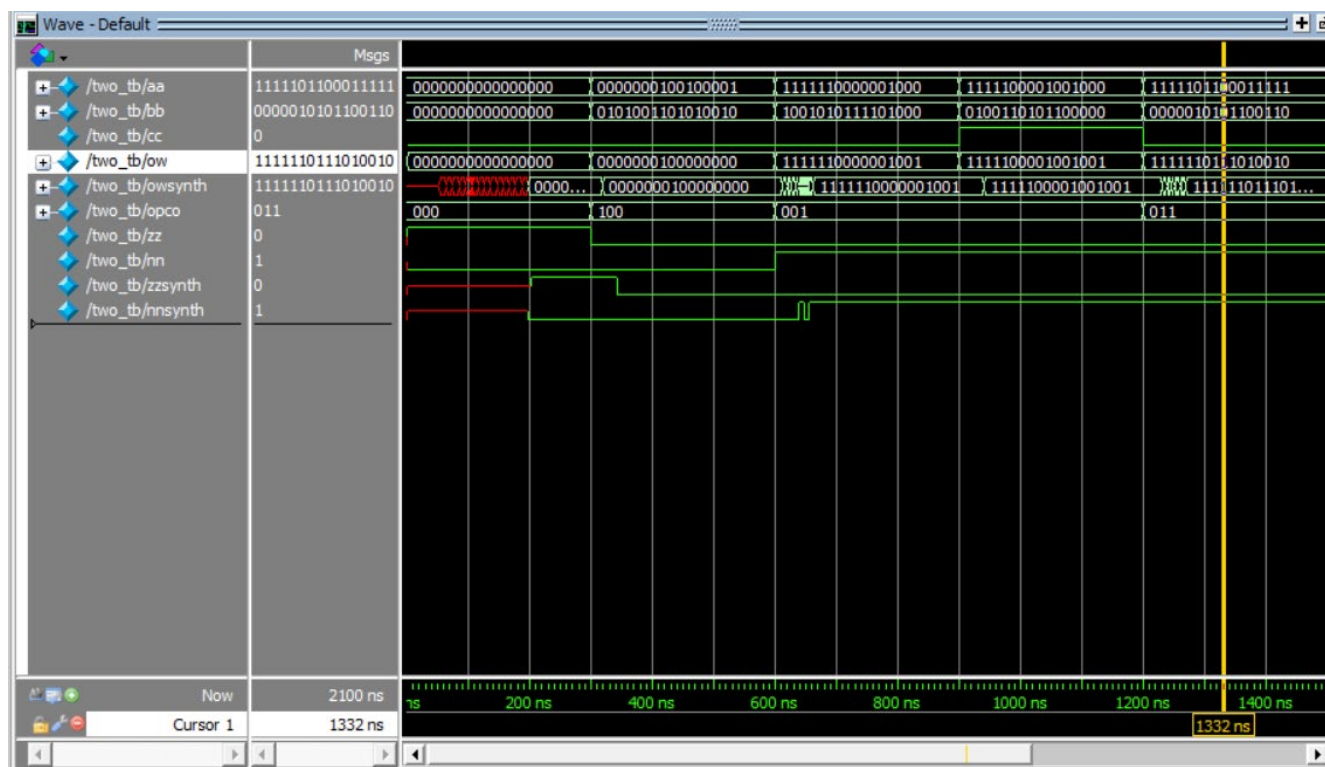
```
=== ALU_struct_3 ===
```

Number of wires:	1005
Number of wire bits:	1052
Number of public wires:	7
Number of public wire bits:	54
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	512
NAND	234
NOR	194
NOT	84

که جمعا از 512 گیت استفاده میکند که همچنان میزان خوبی از گیت های حالت معمولی کاسته ولی رویکرد دوم همچنان به دلیل مصرف کمتر گیت ها اولویت بالاتری دارد. کد این بخش نیز در فایل 2-3.sv قابل مشاهده است.

بخش c)

حال که از میان رویکرد های متفاوت همان رویکرد دوم را انتخاب کردیم، بیایید سری به مدل سیم بزنییم تا نتایج را روی این ساختار نیز امتحان کنیم (لاجیک بخش a و c در این عکس قابل مشاهده است که لاجیک های مربوط به ساختار سنتز شده با پسوند synth مشخص شده اند):



قسمت d) همانگونه که مشاهده میشود، قسمت a بر خلاف قسمت c دیلی نداشته و در واقع هاردوری در آن استفاده نشده است، ولی قسمت c دارای دیلی قابل توجهی است که در مورد آن در قسمت d سوال قبل صحبت شد (در مورد مقایسه این دیلی ها در سوال سه توضیح داده خواهد شد). هر دو این قسمت ها دارای پاسخ درستی به ورودی های متفاوت می باشند که پاسخ قسمت c با تاخیر به پاسخ قسمت a میرسد. در خروجی های یک بیتی مدار سنتز شده، گاهی هازارد مشاهده میشود.

سوال سه

برای بررسی دوباره عکس ها و نمودار ها میتوانید به بخش های قبل مراجعه کنید. بدیهتا ساختار طراحی شده در هر دو سوال دارای جواب صحیح و منطقی می باشند، لیکن بحث سر بهینگی هر کدام خواهد بود. مدار طراحی شده در سوال دو از لحاظ تعداد گیت کمتر بهینه تر محسوب میشود، چرا که با کمتر از سه چهارم تعداد گیت های استفاده شده در مدار سوال اول به همان جواب خواهد رسید. اما از لحاظ تایمینگ، از آنجایی که در مدار قسمت دوم ما باز استفاده ساختار های تکراری را داریم، برای چهار آپکود ابتدایی حداقل یک لول لاجیک نسبت به مدار سوال اول اضافه شده است و در نتیجه این مدار دارای ورست کیس دیلی بیشتری نسبت به مدار سوال اول خواهد بود (تعداد بیشتر لول های لاجیک در دیاگرام های کشیده شده نیز قابل مشاهده است). برای مقایسه، به زمان درایو شدن اولین خروجی در شروع شبیه سازی دقت کنید (در ساختار گیت، از آنجایی که برای داشتن خروجی باید همه گیت های قبلی مقدار غیر x درایو کنند، مقدار دیلی شروع برنامه با مقدار ورست کیس دیلی برابر است). برای مدار سوال اول این زمان حدودا 130 نانو ثانیه است، در حالی که برای مدار سوال دوم این زمان به 200 نانو ثانیه میرسد. بنابراین در هنگام انتخاب هر کدام از این دو ساختار باید توجه کنید که چه نیازی دارید، اگر تعداد گیت برایتان مهم است ساختار دوم و اگر بحث تایمینگ مطرح می باشد، ساختار اول مد نظر است.