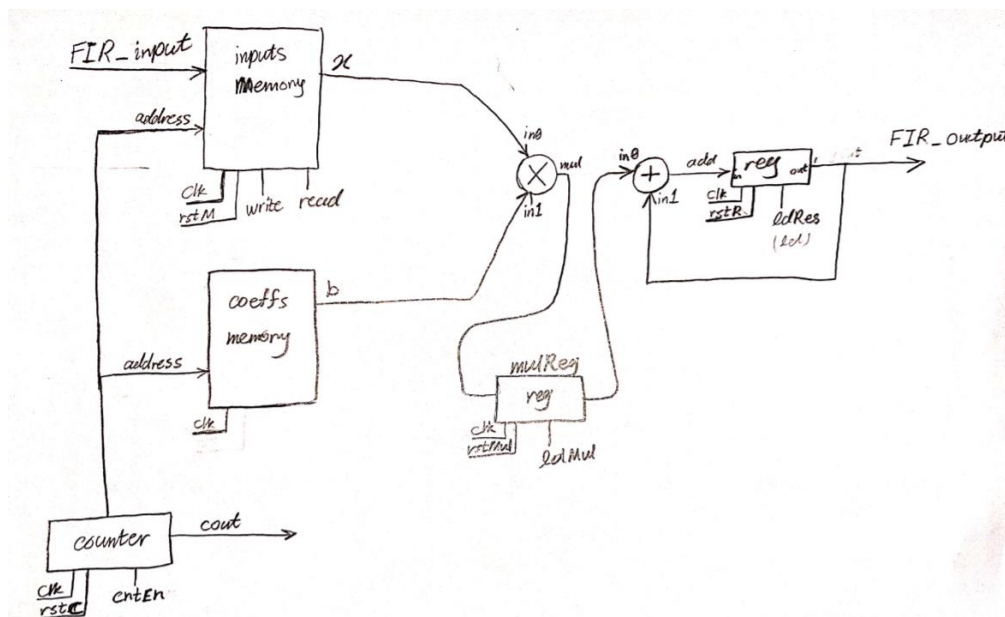


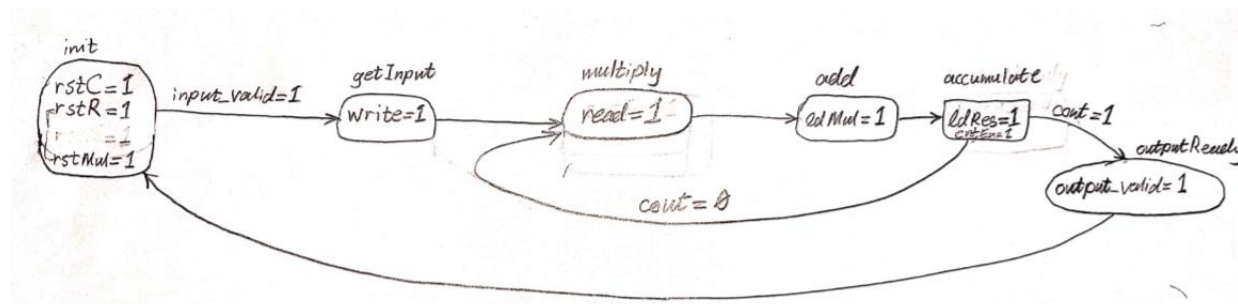
در این فایل گزارش بخش های ذکر شده در فایل تمرین خدمت شما ارائه میشود:

## 1. طراحی سخت افزار

دو تصویر زیر، مبنای طراحی مسیره داده و واحد کنترل این تمرین هستند:



مسیره داده



واحد کنترل

کد نوشته شده برای این بخش به صورت ماژولار بوده و هر یک از کامپوننت هایی که در مسیر داده مشاهده میکنید دارای ماژول جداگانه ای بوده که در رده های بالاتر از آن اینستنس یا اینستنس هایی (مانند رجیستر که در هر دو از یک ماژول اینستنس گرفته شده اند) گرفته شده است.

برای افزایش فرکانس کاری مدار، رجیستری بین ضرب کننده و جمع کننده اضافه شده تا propagation delay مربوط به مجموع تاخیرهای ضرب کننده و جمع کننده تبدیل شود به ماکسیمم تاخیر ضرب کننده و جمع کننده که در این صورت دوره تناوب سیکل ساعت کاهش پیدا میکند و در نتیجه فرکانس کاری مدار (که مدار به طور صحیحی کار میکند) افزایش میابد.

نکته ی دیگری که لازم است ذکر شود اینست که برای در نظر گرفتن out width لازم است به این نکته توجه شود که مجموع یک سری اعداد، تعداد بیت های لازم بخش اعشاری را تغییر نمیدهد و تنها تاثیرش را روی تعداد بیت های لازم برای بخش صحیح میگذارد. به طور مثال اگر تعداد ضرایب (length) در نظر گرفته شده برای فیلتر 64 تا باشد، چون در هر کانولوشن 64 جمع انجام میشود، با فرض اینکه ورودی ها بین -1 و 1 (نابرابر با 1) هستند، حاصل کانولوشن بین -64 و 64 (نابرابر با 64) خواهد بود. لذا

تعداد بیت های لازم برای این مثال به گونه ای باید باشد که در بخش صحیح نتیجه ی نهایی بتوان 63 و 64- را با آن نمایش داد که در نمایش مکمل دو به شرح زیر است:

$$-64 = (10000000)$$

$$63 = 01111111$$

لذا برای حالتی که اندازه فیلتر 64 است و ورودی به صورت  $\text{fix} <16,15>$  است، خروجی به صورت  $\text{fix} <37,30>$  میتواند مناسب باشد.

## 2. توصیف به کمک Verilog

در مدار بالا، تطابق بین عرض ورودی و خروجی در بخش ضرب کننده به صورت ذاتی گنجانده شده (signExtend) که کد آن را در شکل زیر مشاهده میکنید:

```
wire[outWidth-2*inWidth-1:0] signExtend;
wire signed[2*inWidth-1:0] result;
assign result=in0*in1;
assign signExtend=(result[2*inWidth-1]) ? -1 : 0;
assign out={signExtend,result};
```

مموری مربوط به داده های ورودی به صورت FIFO است و هر داده ای که وارد میشود به خانه صفر حافظه وارد میشود و با هربار ورود داده جدید، داده های قبلی به آدرس بعدی منتقل میشوند، لذا با این ترفند برای انجام کانولوشن میتوان برای هر دو مموری داده های ورودی و ضرایب، یک شمارنده واحد برای ارجاع به آدرس ها به کار برد:

```
always @(posedge clk,posedge rst)
begin
    if(rst)
        for(i=0;i<length;i=i+1)
            memory[i]<=0;
        else if(write)
            begin
                for(i=1;i<length;i=i+1)
                    memory[i]<=memory[i-1];
                memory[0]<=in;
            end
        else if(read)
            out=memory[address];
end
```

خروجی cout وقتی یک میشود که تمام عملیات کانولوشن به طور کامل تمام شده (یعنی آدرس های 0 تا  $\text{length}-1$  هر دو مموری در هم ضرب و با هم جمع شده اند) و نشان دهنده ی پایان عملیات است و این موضوع به واحد کنترل اطلاع داده میشود:

```
assign address=cntReg;
assign cout=(address>=length-1) ? 1 : 0;

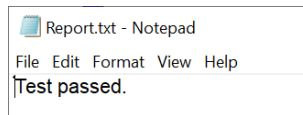
always@(posedge clk,posedge rst)
begin
    if(rst)
        cntReg=0;
    else if(cntEn)
        cntReg=cntReg+1;
end
```

### 3. درستی سنجی با روش شبیه سازی

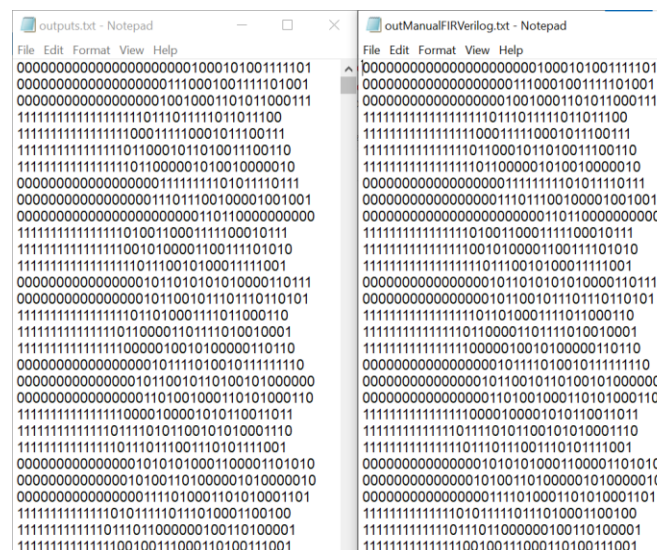
کد زیر که الهام گرفته از کد ارسالی به همراه فایل تمرین است، به این صورت عمل میکند که به تعداد DATA\_LEN از فایل ورودی ها، ورودی خوانده میشود و با هربار ورود داده جدید محاسبات انجام میشود و داده ی بعدی پس از صدور سیگنال output\_valid وارد فیلتر میشود و این روند ادامه میابد. نتایج مربوط به مقایسه داده حاصل از فیلتر طراحی شده و خروجی مرجع هم در کنسول نمایش داده میشود و هم در دو فایل Report.txt (که نتایج خطا و اتمام تست بنچ در آن فایل گزارش میشود) و outManualFIRVerilo.txt (که خروجی های فیلتر ما در آن ذخیره میشود) پس از شبیه سازی قابل مشاهده هستند.

```
initial
begin
    fp = $fopen("outManualFIRVerilog.txt");
    fpReport = $fopen("Report.txt");
    #40;
    $display("Testing %d Samples...",DATA_LEN);
    for(i = 0; i < DATA_LEN; i = i + 1)
    begin
        input_valid=1;
        din=input_data[i];
        @(posedge output_valid)
        begin
            input_valid=0;
            $fwrite(fp,"%b\n",dout);
            temp_out=expected_data[i];
            if(dout != temp_out[OUT_WIDTH-1:0])
            begin
                $display("test failed: %d  input: %x expected: %x output: %x" , i, din, temp_out[OUT_WIDTH-1:0], dout);
                $fwrite(fpReport,"test failed: %d  input: %x expected: %x output: %x\n" , i, din, temp_out[OUT_WIDTH-1:0], dout);
            end
        end
    end
    $fwrite(fpReport,"Test passed.\n");
    $fclose(fpReport);
    $fclose(fp);
    $display("Test Passed.");
    $finish;
end
```

تصاویر خروجی حاصل از 15000 ورودی را در شکل زیر مشاهده میکنید:



نشان دهنده ی اینست که هیچ اختلافی بین خروجی فیلتر و خروجی مرجع وجود نداشته



عدم اختلاف بخشی از خروجی ها را در شکل بالا میبینید

## 5. سنتز

در سنتز های انجام شده فرض شده است که ورودی ها مقداری در بازه ی (1,-1] دارند. لذا مانند توضیحات ارائه شده در بخش اول این فایل، فرمت عرض داده خروجی برای حالت های مختلف به شرح زیر است:

- 1) in width=fix<8,7> and length=50 => out width = fix<21,14>
- 2) in width=fix<8,7> and length=100 => out width = fix<22,14>
- 3) in width=fix<16,15> and length=50 => out width = fix<37,30>
- 4) in width=fix<8,7> and length=100 => out width = fix<38,30>

در ادامه نتایج حاصل از سنتز چهار حالت فوق را مشاهده میکنید:

1. عرض ورودی 8 بیت و اندازه 50 و عرض خروجی 21 بیت:

```
module FIR#(parameter IN_WIDTH=8,OUT_WIDTH=21,LENGTH=50) (
    clk,
    rst,
    FIR_input,
    input_valid,
    FIR_output,
    output_valid
);
```

تغییرات اعمال شده روی پارامترهای top level module

Flow Summary	
Flow Status	Successful - Tue Oct 19 21:55:51 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FIR
Top-level Entity Name	FIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	560 / 33,216 ( 2 % )
Total combinational functions	346 / 33,216 ( 1 % )
Dedicated logic registers	459 / 33,216 ( 1 % )
Total registers	459
Total pins	33 / 475 ( 7 % )
Total virtual pins	0
Total memory bits	0 / 483,840 ( 0 % )
Embedded Multiplier 9-bit elements	1 / 70 ( 1 % )
Total PLLs	0 / 4 ( 0 % )

خلاصه گزارش سنتز

Slow Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	197.12 MHz	197.12 MHz	clk	

حداکثر فرکانس کاری مدار در این حالت

2. عرض ورودی 8 بیت و اندازه 100 و عرض خروجی 22 بیت:

```
module FIR#(parameter IN_WIDTH=8,OUT_WIDTH=22,LENGTH=100) (
    clk,
    rst,
    FIR_input,
    input_valid,
    FIR_output,
    output_valid
);
```

تغییرات اعمال شده روی پارامترهای top level module

Flow Summary	
Flow Status	Successful - Tue Oct 19 22:07:06 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FIR
Top-level Entity Name	FIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	994 / 33,216 ( 3 % )
Total combinational functions	570 / 33,216 ( 2 % )
Dedicated logic registers	845 / 33,216 ( 3 % )
Total registers	845
Total pins	34 / 475 ( 7 % )
Total virtual pins	0
Total memory bits	800 / 483,840 ( < 1 % )
Embedded Multiplier 9-bit elements	1 / 70 ( 1 % )
Total PLLs	0 / 4 ( 0 % )

خلاصه گزارش سنتز

	Fmax	Restricted Fmax	Clock Name	Note
1	140.39 MHz	140.39 MHz	clk	

حداکثر فرکانس کاری مدار در این حالت

3. عرض ورودی 16 بیت و اندازه 50 و عرض خروجی 37 بیت:

```
module FIR#(parameter IN_WIDTH=16,OUT_WIDTH=37,LENGTH=50) (
    clk,
    rst,
    FIR_input,
    input_valid,
    FIR_output,
    output_valid
);
```

تغییرات اعمال شده روی پارامترهای top level module

Flow Summary	
Flow Status	Successful - Tue Oct 19 22:21:27 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FIR
Top-level Entity Name	FIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	999 / 33,216 ( 3 % )
Total combinational functions	584 / 33,216 ( 2 % )
Dedicated logic registers	867 / 33,216 ( 3 % )
Total registers	867
Total pins	57 / 475 ( 12 % )
Total virtual pins	0
Total memory bits	800 / 483,840 ( < 1 % )
Embedded Multiplier 9-bit elements	2 / 70 ( 3 % )
Total PLLs	0 / 4 ( 0 % )

خلاصه گزارش سنتز

Slow Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	134.01 MHz	134.01 MHz	clk	

حداکثر فرکانس کاری مدار در این حالت

4. عرض ورودی 16 بیت و اندازه 100 و عرض خروجی 38 بیت:

```
module FIR#(parameter IN_WIDTH=16,OUT_WIDTH=38,LENGTH=100) (
    clk,
    rst,
    FIR_input,
    input_valid,
    FIR_output,
    output_valid
);
```

تغییرات اعمال شده روی پارامترهای top level module

Flow Summary	
Flow Status	Successful - Tue Oct 19 22:28:30 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FIR
Top-level Entity Name	FIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	1,926 / 33,216 ( 6 % )
Total combinational functions	1,114 / 33,216 ( 3 % )
Dedicated logic registers	1,669 / 33,216 ( 5 % )
Total registers	1669
Total pins	58 / 475 ( 12 % )
Total virtual pins	0
Total memory bits	1,600 / 483,840 ( < 1 % )
Embedded Multiplier 9-bit elements	2 / 70 ( 3 % )
Total PLLs	0 / 4 ( 0 % )

خلاصه گزارش سنتز

Slow Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	132.59 MHz	132.59 MHz	clk	

حداکثر فرکانس کاری مدار در این حالت

در جدول زیر خلاصه عملکرد مدار را در کوارتس مشاهده میکنید:

عرض ورودی	اندازه	عرض خروجی	تعداد المان های منطقی	تعداد فلیپ ها	حداکثر فرکانس کاری
8	50	21	560	459	197.12 MHz
8	100	22	994	845	140.39 MHz
16	50	37	999	867	134.01 MHz
16	100	38	1926	1669	132.59 MHz

## توضیحات تکمیلی

در تنظیمات کوارتس، به منظور افزایش فرکانس، تنظیمات مربوط به بهینه سازی را از حالت balance به speed تغییر دادم تا در هنگام سنتز اولویت با افزایش سرعت مدار باشد. همچنین Fitter Effort را به حالت استاندارد تغییر دادم و همچنین ALM Register Packing Effort را به Low تغییر دادم.

همچنین چگونگی پیدا کردن و اضافه کردن ضرب کننده پایپ لاین را از مسیر

Tools -> MegaWizard Plug-In Manager

فراگرفتم و میتوانم با استفاده از ضرب کننده ی LPM\_MULT که قابلیت پایپ لاین کردن دارد، سرعت پردازش را افزایش دهم. اما با توجه به کمبود وقت و نیاز به وقت کافی جهت اعمال تغییرات در بخش های مختلف طراحی، در حال حاضر انجام این کار برای بنده مقدور نبوده. فایل های multip.v و multip\_bb.v مربوط به همین ضرب کننده هستند که در پروژه ام از آن ها استفاده نکردم اما میتوانید در پوشه ی CA 1 آن را در کنار سایر فایل های ورپلاگ پروژه مشاهده کنید.

فایل های کوارتس هم در پوشه ی کوارتس قرار دارند.