

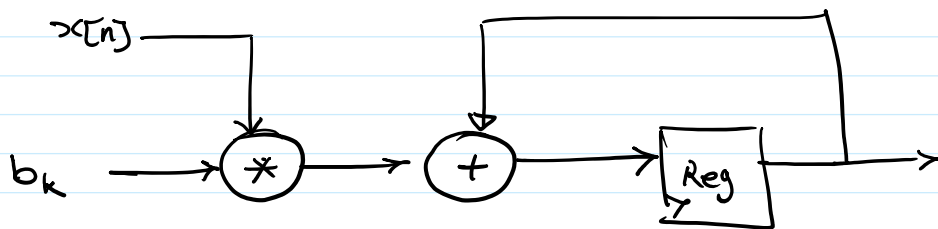
سید محمد حسینی ۸۵۱۵۹۴۵۶۱ مورخه ۱۳۹۷/۰۹/۰۱

در این پروژه به استناد از طراحی سخت افزار یک FIR فیلتر با بهره سازی می کنیم.

در ابتدا می نمود که در دستور کار لایه شده است، سیستم را به توجه به مدل طراحی می کنیم:

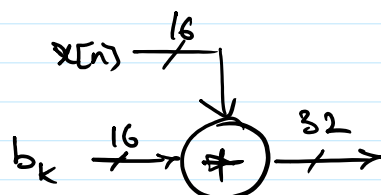
$$y[n] = \sum_{k=0}^M k[k] \cdot x[n-k]$$

چون نمی توان  $M$  را قبل تعیین کرد، ما به صورت بازتری سیستم را طراحی می کنیم:



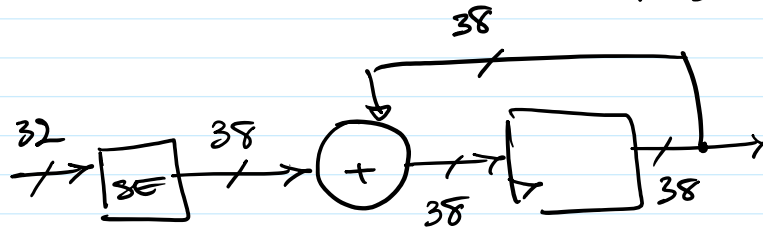
سیستم طراحی شده در بالا یک سیکل محاسباتی است، به علاوه محاسبه خروجی بیکر جسته می شود تا با محاسبات بعدی برای یک  $y[n]$  نتیجه دلی شود. برای این مثال برای محاسبه یک  $y[n]$  لازم است که ۶۴ بار محاسبات انجام شود و ما به آن ۶۴ بار آسان می گوییم.

می نمود که معلوم است ورودی که ۱۶ بیت بوده و فرایب عم ۱۶ بیت هسته پس عمل خوب خروجی اش ۳۲ بیت است:



۶۴ فریب داریم، پس جمع ۶۴ بار آسان می گوییم، پس اگر ۶۴ بار عددی را جمع کنیم آن عدد در ۸ فریب کرده ایم یا  $(N)$  و اگر آن را نسبت داده ایم. پس خروجی جمع کننده با توجه به مقدار

فیلتر کرده ایم یا (N) و برای راسیت داده ایم. پس خروجی جمع کننده عبارتست به مقدار  
خواب باید  $32+4$  بیت بود :



+ چون تعداد بیت در این سیگنال 38 بیت است و carry 2 بیت است.

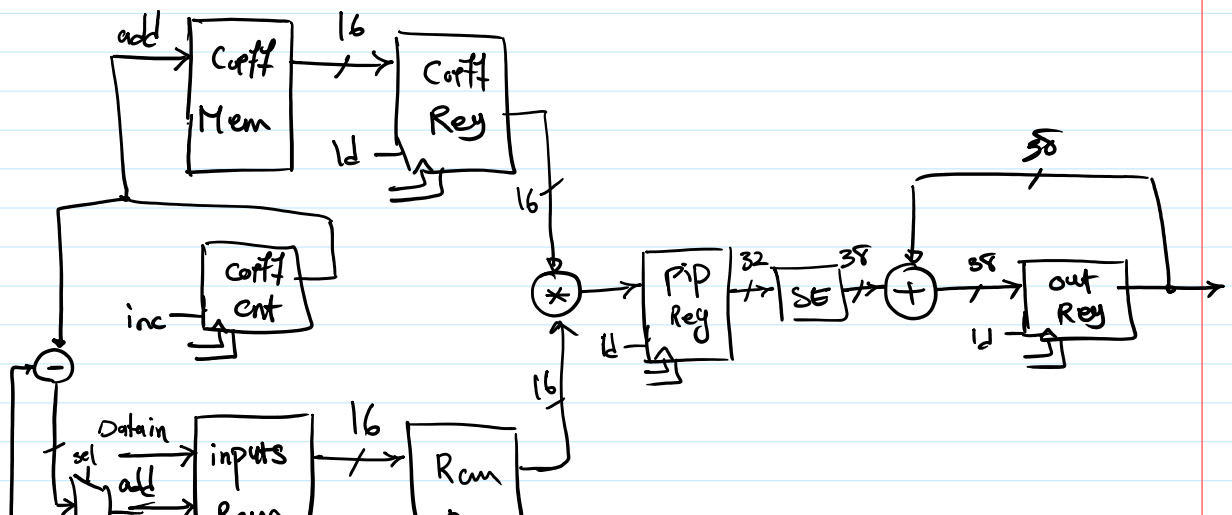
+ تمامی حسابات براس Fix point صورت گرفته است.

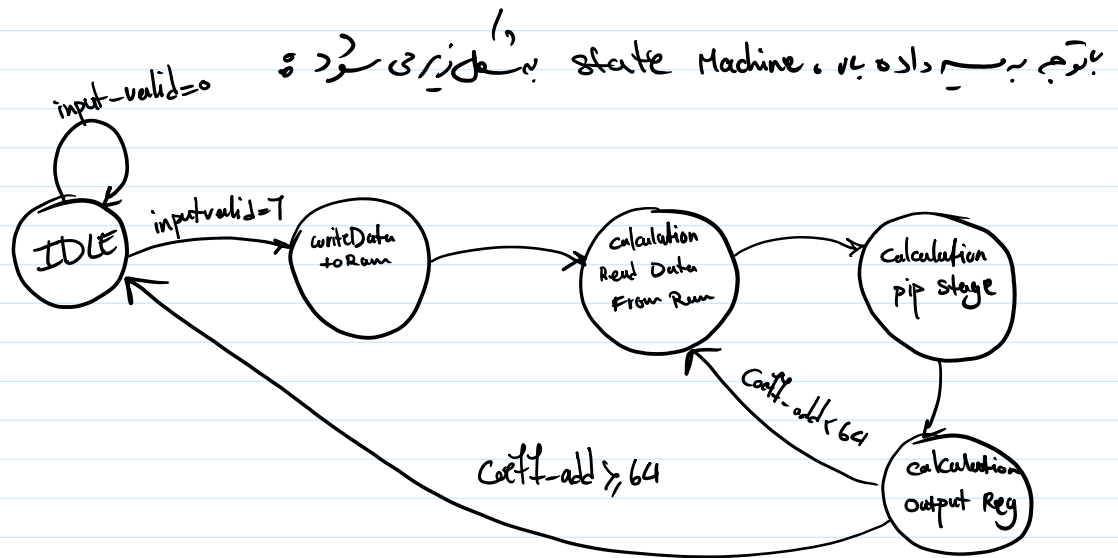
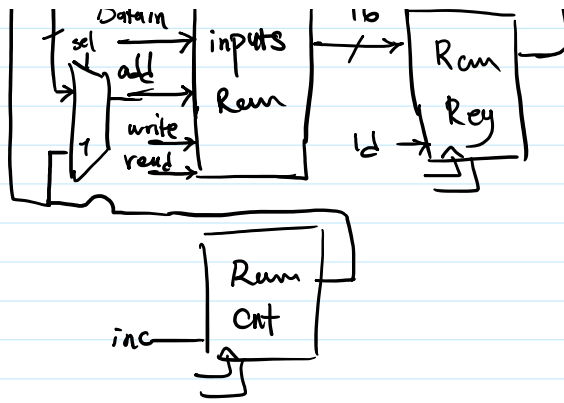
۱) برای محاسبه هر [N] به 44 ورودی نیاز داریم. برای همین مقول لازم است 44 داده پس  
راسته دایک کنیم. برای اینکار من یک Memory به اندازه کل ورودی 4 در داخل FIR قرار دادم  
که هر ورودی که می آید داخل رم نه دایک می کنم.

۲) برای خوابید Memory قرار دادم که ورودی اگر در هنگام ساخته شدن اینک فیل می خواند  
و ذخیره می کند.

۳) برای آدرس input Ram, coeff memory, 2 کانترا دلفن می دهیم که آدرس را  
در خروجی ذخیره می کند.

۴) برای pipeline یک رجیتر بین فیلتر کننده جمع کننده قرار می دهیم  
می دهیم به شکل زیر است :





باتوجه به SM بالا، ابتدا مدلهما  $input\_valid$  می ماند بعد از آن سیکل که مدار شروع می شود. ابتدا داده داخل  $input\_Ram$  ذخیره می شود. سپس داده آرایری  $Ram$  و  $Cofft$  خوانده می شوند. (چون رجیستر بعد از مدتی آبی باشد، یک سیکل طول فراوان می کشد) پس فرستاده می شود به  $Calculation$  و  $Reg$   $pip$  ذخیره می شود و بعد از آن هم عملیات جمع انجام می شود.

برای ارزیابی مدار، باتوجه به Testbench که در فایل  $testbench$  نوشته شده به ازای تمامی ورودی ها، خروجی را با خروجی داده شده مقایسه می کند و در صورت مغایرت یک سیکل ذخیره می کند. در ابتدا تمامی داده ها اشتباه بودند و بعد از رفع ایرادات مدار به درستی عمل می کرد. خروجی که را این بار در در فایل  $outManualFIRvalinby$  ذخیره کرد. نتیجه سی دی در صورت ۱۰۰٪ قبول شد و خروجی که بعد از مدت طولانی در فایل نوشته شد.

```
# Start Testing FIR Filter..
# Testing      221184 Samples...
# Input Width :      16, Output Width :      38
# Test Passed.
# Break in Module FIR_TB at E:/UT/fir2/FIRFilter_TB.v line 101
```

د- نت: :

همانگونه در صورت پررنگ شده است، برای طولی ۸،۱۴ و اندازه ۵۰،۱۰۰  
نت را انجام دادیم. خروجی که مشخص بود:

Length	Width	Total logic Elements	Column1	Total	Total memory bits	Embedded Multiplier 9-bit elements
		Total combinational functions	Dedicated logic registers			
50	8	408 / 33,216 ( 1 % )	450 / 33,216 ( 1 % )	450	520 / 483,840 ( < 1 % )	1 / 70 ( 1 % )
100	8	736 / 33,216 ( 2 % )	851 / 33,216 ( 3 % )	851	520 / 483,840 ( < 1 % )	1 / 70 ( 1 % )
50	16	704 / 33,216 ( 2 % )	874 / 33,216 ( 3 % )	874	1,040 / 483,840 ( < 1 % )	2 / 70 ( 3 % )
100	16	1,320 / 33,216 ( 4 % )	1,675 / 33,216 ( 5 % )	1675	1,040 / 483,840 ( < 1 % )	2 / 70 ( 3 % )

همانگونه انتظاری رفت، چون غربال شده ۸ بیت است زمانه داده ۱۲ بیت می شود، ۲ ضرب  
کننده است و می شود. همچنین با افزایش اندازه، تعداد رجیسترهای بیشتری لازم است.  
\* مگر می گویید که در داخل فلدر photos قرار دارند.

جمع بندی: با استفاده از پایه سازی سخت افزاری توانستیم یک فلتر FIR بسازیم. برنامه  
نویسه شده قابلیت تغییر با هر اندازه و طول را دارد چون با استفاده از parameter  
همی زیر مارول گرفته شده اند. بعد از تهیه سازی تمامی خروجی که به درستی دریافت  
شدند و به سطر کردن برنامه برای ۴ حالت توانستیم میزان سخت افزار مورد استفاده  
را بدست آوریم.