



دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر

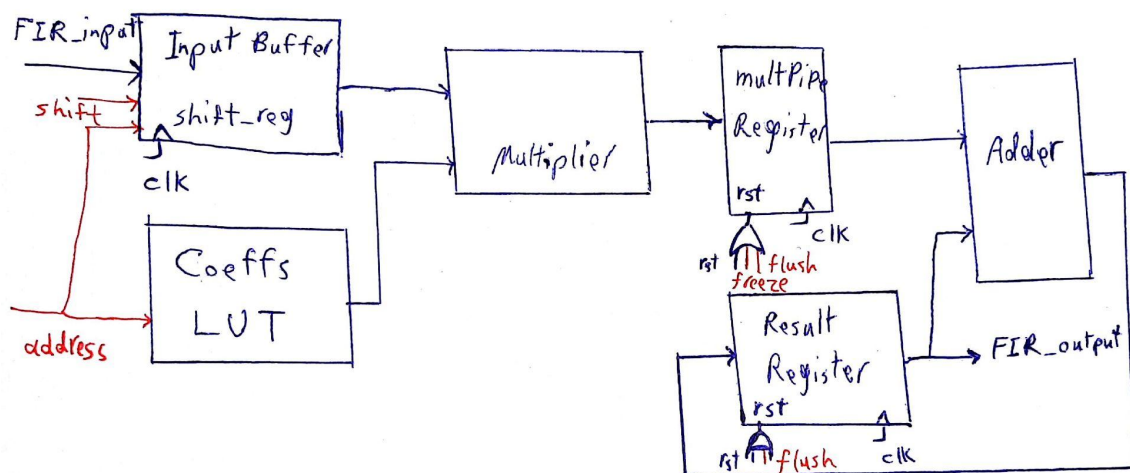
درس طراحی سیستم‌های نهفته مبتنی بر FPGA

تکلیف کامپیوتری 1

امیرعلی آرزم جو - 810198342

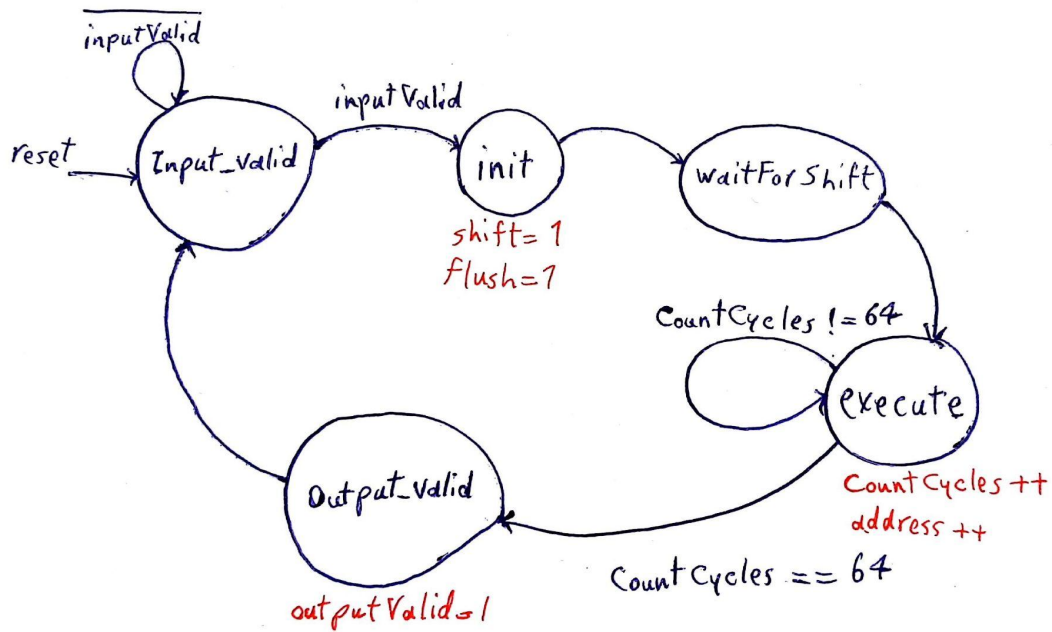
گام 1: طراحی سخت افزاری

برای دریافت ورودی، به یک buffer مناسب نیاز داریم که برای تولید هر output یک input را دریافت کند و 63 input قبلی را نیز در یک صف نگه دارد، پس به یک shift register-file نیاز داریم، نام این ماژول shift_reg.sv است؛
برای ذخیره ی coefficient ها از یک Look-up table با سایز 64 استفاده می شود، برای این بخش ماژول جداگانه نیازی نیست و از یک آرایه logic استفاده می گردد؛
هر input در coefficient متناظر ضرب میشود و با مقادیر قبلی جمع میشود، پس یک register، یک adder و یک multiplier هم مورد نیاز است؛
برای پایپ لاین کردن مسیر داده یک رجیستر دیگر هم نیاز است.



شکل 1: مسیر داده

کنترلر نیز به صورت state machine طراحی شد.



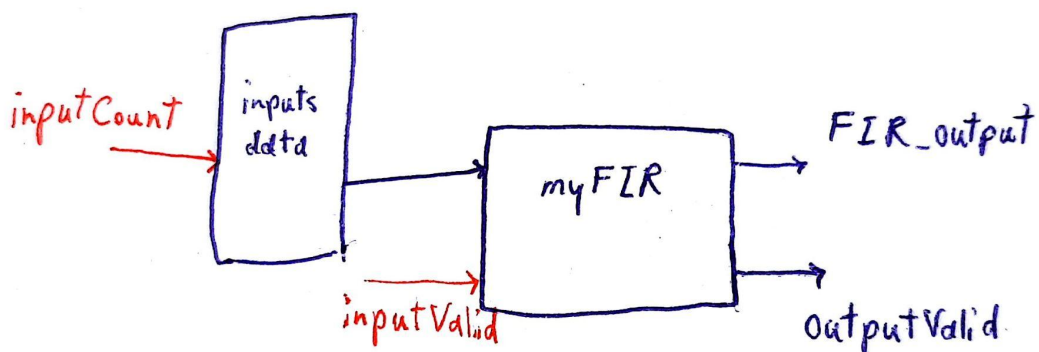
شکل 2: دیاگرام حالت کنترلر

گام 2: توصیف به کمک Verilog - لینک Github [اینجا](#)

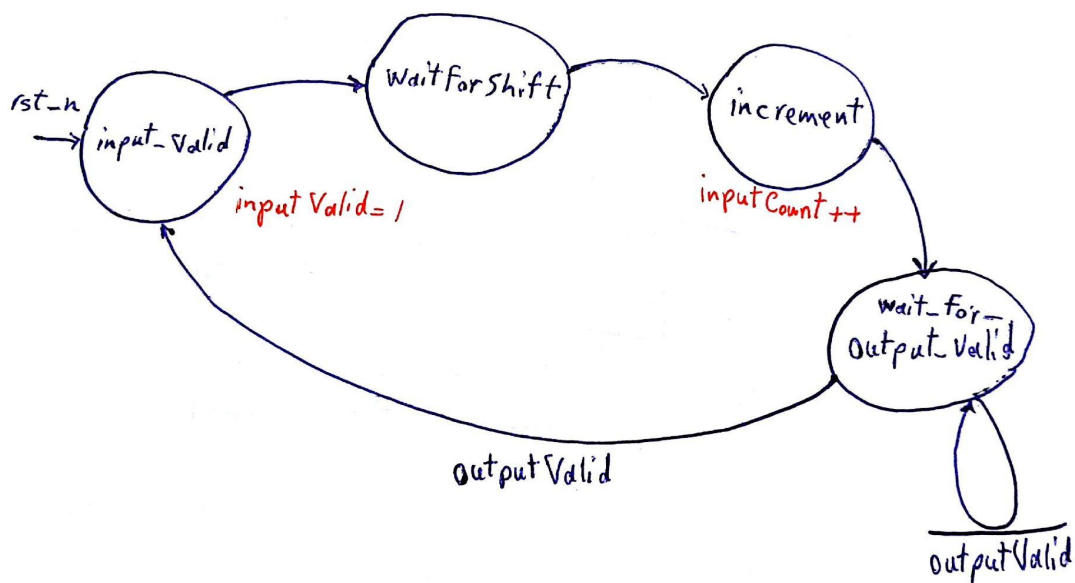
ماژول ها با فرمت SystemVerilog و ماژول اصلی با اسم myFIR.sv به پروژه اضافه شد.

گام 3: درستی سنجی با روش شبیه سازی

مقادیر داخل فایل inputs.txt مانند کد نمونه ی موجود در صورت پروژه خوانده شده و در مموری input_data ذخیره می شود.

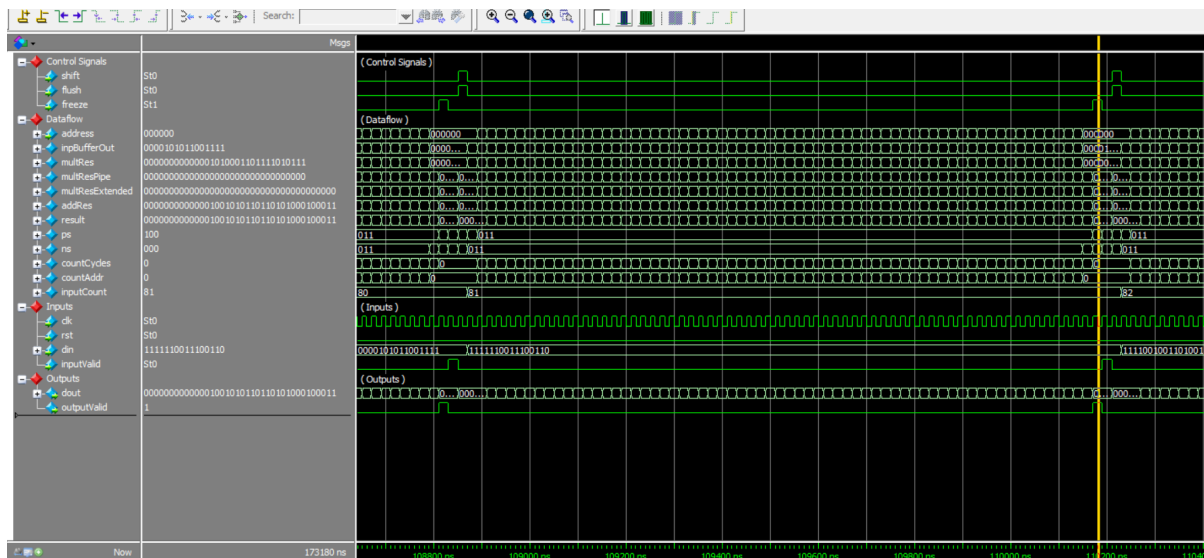


شکل 3: نحوه اتصال ماژول اصلی در testbench



شکل 4: دیاگرام حالت کنترلر testbench

با استفاده از modelsim و بررسی موج خروجی های بدست آمده و مقایسه با مقادیر داده شده در outputs.txt عملکرد طراحی تایید می گردد.



شکل 5: شکل موج شبیه سازی testbench و مشاهده ی 81 امین output در سیگنال dout که با مقدار مورد انتظار تطابق دارد.

گام 4: درستی سنجی با روش assertion

برای جلوگیری از قرار گرفتن assertion ها و کد RTL کنار همدیگر که در سنتز مشکل ایجاد می کنند، همه ی assertion ها در انتهای testbench قرار داده شدند.

برای درستی سنجی بخش های محاسباتی Adder و Multiplier دو property می نویسیم و یک شرط disable iff نیز قرار می دهیم تا پس از pass شدن تعداد دلخواهی، assertion متوقف شود و terminal شلوغ نشود.

```
property checkMult;
    @(posedge clk) disable iff(inputCount > 4) (1'b1==1'b1)
|->((uut1.uut1.mult1.a * uut1.uut1.mult1.b) == uut1.uut1.mult1.res);
endproperty

checkingMult: assert property (checkMult)
$display($stime,,, "\t\tMult PASS"); else $display($stime,,, "\t\tMult
FAIL");

property checkAdd;
    @(posedge clk) disable iff(inputCount > 4) (1'b1==1'b1)
|->((uut1.uut1.add1.a + uut1.uut1.add1.b) == uut1.uut1.add1.res);
endproperty

checkingAdd: assert property (checkAdd) $display($stime,,, "\t\tAdd
PASS"); else $display($stime,,, "\t\tAdd FAIL");
```

برای بررسی اینکه آیا Counter آدرس LUT را در حالت execute می شمارد یا خیر هم یک property تشکیل می دهیم.

```
property checkCtrlAddr;
    @(posedge clk) disable iff(inputCount > 6) (uut1.uut2.ps == 3)
|-> ($past(uut1.uut2.countAddr) == (uut1.uut2.countAddr-1));
endproperty

checkingCtrlAddr: assert property (checkCtrlAddr)
$display($stime,,, "\t\tCounter is counting!");
else $display($stime,,, "\t\tCounter isn't counting");
```

مهمتر از همه، مقادیر فایل output.txt را در یک آرایه ی حافظه ذخیره می کنیم و در یک property هر output تولید شده توسط ماژول را با مقادیر آرایه مطابقت می دهیم، انتظار داریم به ازای همه ی مقادیر pass شود:

```
property outVal;
    @(posedge clk) $rose(outputValid) |-> (temp_out ==
expected_outputs[inputCount-1]);
endproperty

checkOut: assert property (outVal) $display($stime,,, "\t\tOut
PASS"); else $display($stime,,, "\t\tOutput FAIL");
```

یک property برای بررسی عملکرد state machine بخش کنترلر می نویسیم:

```

sequence stateSeq;
    (uut1.uut2.ps == 1) ##1 (uut1.uut2.ps == 2) ##1 (uut1.uut2.ps
== 3) [*64];
endsequence
property inpVal;
    @(posedge clk) disable iff(inputCount > 40) $rose(inputValid)
|=> stateSeq;
endproperty
checkStates: assert property (inpVal)
$display($stime,,, "\t\tController PASS"); else
$display($stime,,, "\t\tController FAIL");

```

```

#      1250      Counter is counting!
#      1250      Mult PASS
#      1250      Add PASS
#      1270      Counter is counting!
#      1270      Mult PASS
#      1270      Add PASS
#      1290      Counter is counting!
#      1290      Mult PASS
#      1290      Add PASS
#      1310      Counter is counting!
#      1310      Mult PASS
#      1310      Add PASS
#      1330      Counter is counting!
#      1330      Mult PASS
#      1330      Add PASS
#      1350      Counter is counting!
#      1350      Mult PASS
#      1350      Add PASS
#      1370      Counter isn't counting
#      1370      Controller PASS
#      1370      Mult PASS
#      1370      Add PASS
#      1390      Out PASS

```

شکل 6: مقادیر نشان داده شده در پنجره ی transcript

گام 5: سنتز

برای سنتز باید کد زیر در ماژول datapath را comment out کرد:

```
initial
begin
$readmemb("coeffs.txt", coeffs);
end
```

به ازای width = 50 و size = 100 مقدار Fmax به صورت زیر است:

	Fmax	Restricted Fmax	Clock Name	Note
1	446.23 MHz	420.17 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)

Flow Summary	
Flow Status	Successful - Sat Oct 15 17:33:53 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FPGA_CA1
Top-level Entity Name	myFIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	15 / 33,216 (< 1 %)
Total combinational functions	14 / 33,216 (< 1 %)
Dedicated logic registers	12 / 33,216 (< 1 %)
Total registers	12
Total pins	20 / 475 (4 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

شکل 7: Resources for width = 8 & size = 50

Flow Summary

Flow Status	Successful - Sat Oct 15 17:41:12 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FPGA_CA1
Top-level Entity Name	myFIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	15 / 33,216 (< 1 %)
Total combinational functions	14 / 33,216 (< 1 %)
Dedicated logic registers	12 / 33,216 (< 1 %)
Total registers	12
Total pins	36 / 475 (8 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Resources for width = 16 & size = 50 : شکل 8

Flow Summary

Flow Status	Successful - Sat Oct 15 17:47:52 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FPGA_CA1
Top-level Entity Name	myFIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	16 / 33,216 (< 1 %)
Total combinational functions	15 / 33,216 (< 1 %)
Dedicated logic registers	13 / 33,216 (< 1 %)
Total registers	13
Total pins	20 / 475 (4 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Resources for width = 8 & size = 100 : شکل 9

Flow Summary	
Flow Status	Successful - Sat Oct 15 17:49:41 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	FPGA_CA1
Top-level Entity Name	myFIR
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	16 / 33,216 (< 1 %)
Total combinational functions	15 / 33,216 (< 1 %)
Dedicated logic registers	13 / 33,216 (< 1 %)
Total registers	13
Total pins	36 / 475 (8 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

شکل 10: Resources for width = 16 & size = 100