

Lab. 08

Logic Design Lab.

Spring 2022

Prof. ChangGun Lee

(cglee@snu.ac.kr)

TA. Hayeon Park

TA. Haejoo Jeon

TA. Seohwan Yoo

(rubis.ld.ta@gmail.com)

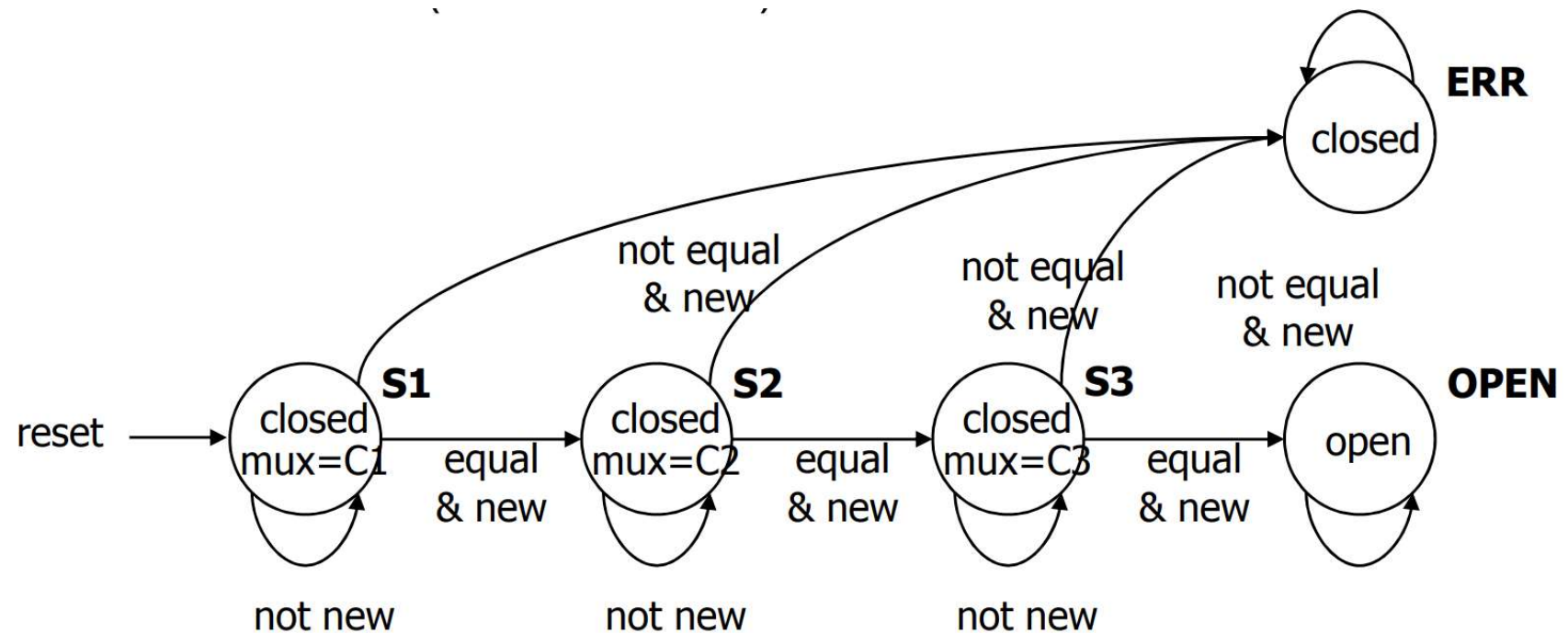
Contents

- **Finite state machine**
- **Moore/Mealy machine**
- **Implementation on Logic Design Board**
- **Vending machine**

Announcement

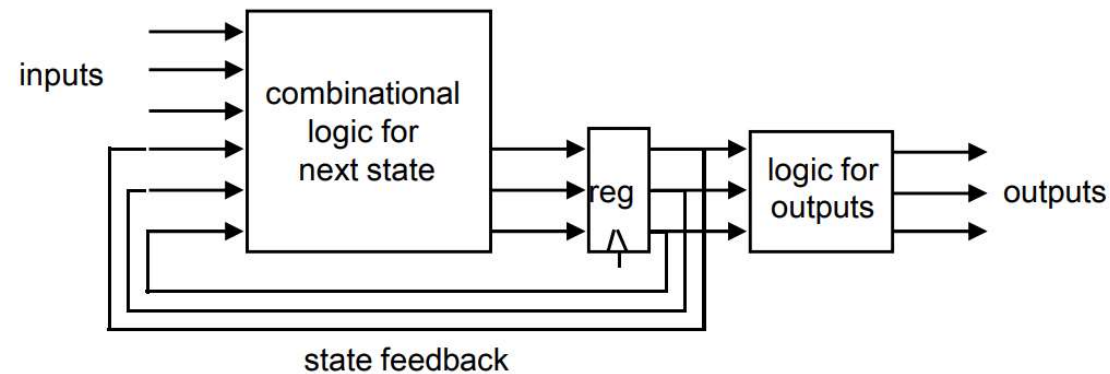
- There will be an announcement of the final project in next lab session.
- Please make sure your board is working correctly

Finite state machine

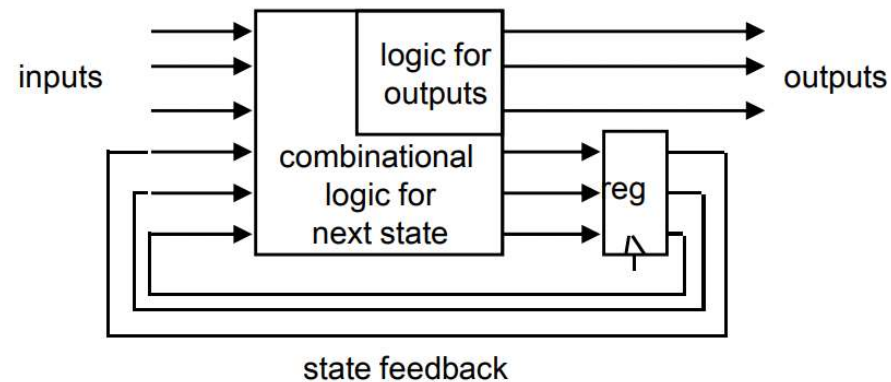


Moore/Mealy machine

- MOORE

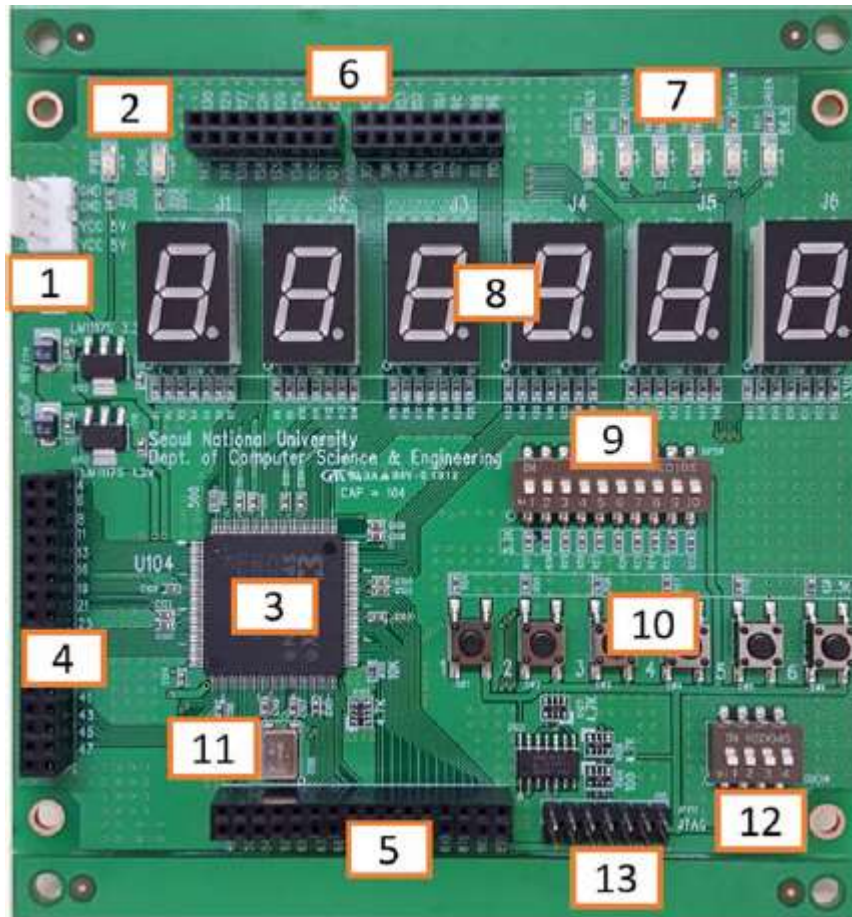


- MEALY



Implementation on Logic Design Board (Review)

SNU Logic Design Board



- ① Power Input : Molex Connector (DC 5V Only)
- ② Power, DONE LED
- ③ FPGA : XC3S50AN-4TQG144C
- ④ User I/O Ports (P1) : 16X2 2.54mm Pitch
- ⑤ User I/O Ports (P2) : 16X2 2.54mm Pitch
- ⑥ User I/O Ports (P3, P4) : 2 8X2 2.54mm Pitch
- ⑦ User Output LEDs : 6 output LEDs
- ⑧ User Output LEDs : 6 7-segment LEDs
- ⑨ User Input Switches : 10pin Dip Switch
- ⑩ User Input Switches : 6 Tactile Switches
- ⑪ Oscillator : 50MHz
- ⑫ Mode Select Switch
- ⑬ JTAG Header

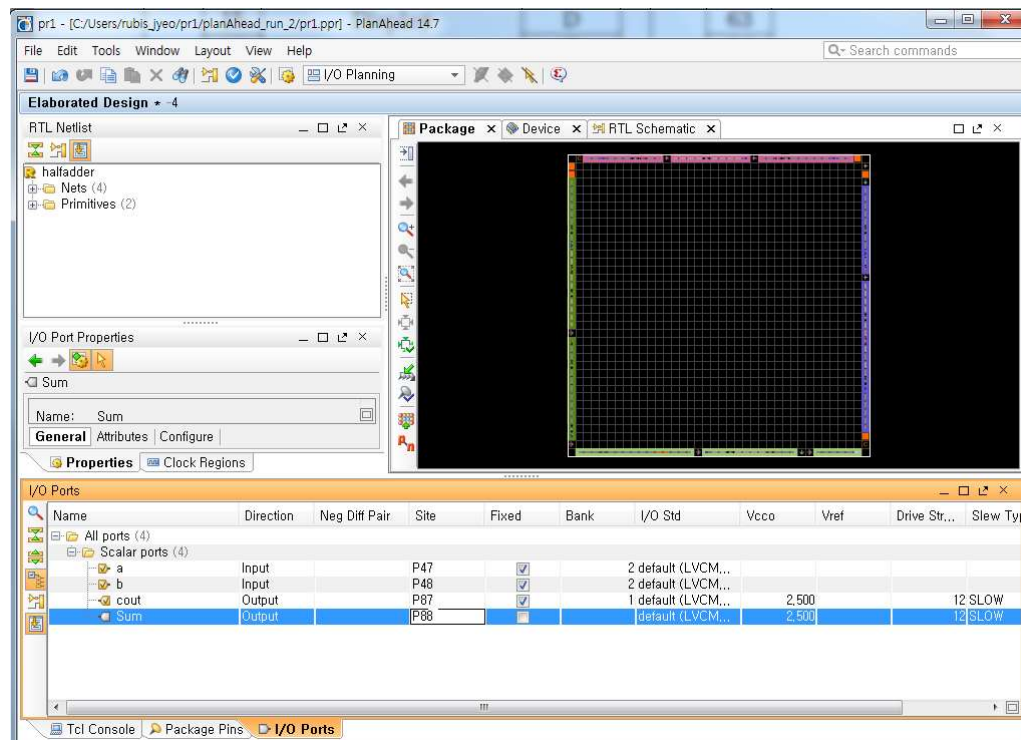
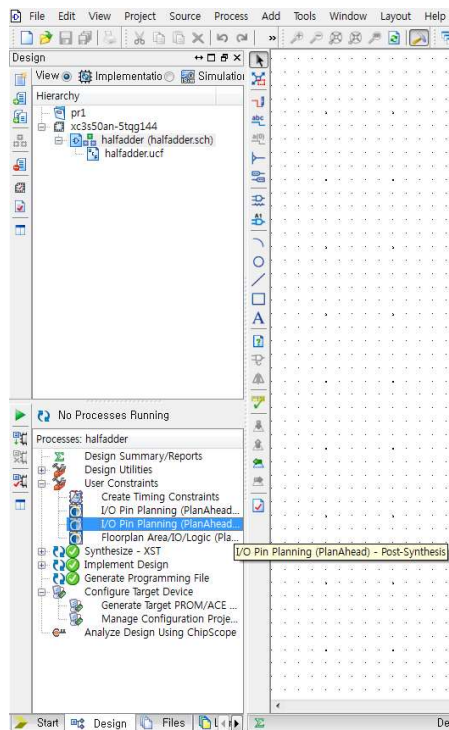
SNU Logic Design Board

- Pin Number Mapping

Pin Num	Component
P1	3
	4
	5
	6
	7
	8
	10
	11
	12
	13
	15
	16
	18
	19
	20
	21
	24
	25
	27
	28
	29
	30
	31
	32
	33
	41
	42
	43
	44
	45
	46
	47
	48
	49
	50
	51
	54
	55
	58
	59
	60
	62
	63
	64
	68
	69
	70
	71
	72
	75
	76
	77
	78
	79
	82
	83
	84
	85
	87
	88
	90
	91
	92
	93
	96
	98
	99
	101
	102
	103
	104
	105
	110
	111
	112
	113
	114
	115
	116
	117
	120
	121
	124
	125
	126
	127
	129
	130
	131
	132
	134
	135
	138
	139
	141
	142
	143
	144
	145
	146
	147
	148
	149
	150
	151
	152
	153
	154
	155
	156
	157
	158
	159
	160
	161
	162
	163
	164
	165
	166
	167
	168
	169
	170
	171
	172
	173
	174
	175
	176
	177
	178
	179
	180
	181
	182
	183
	184
	185
	186
	187
	188
	189
	190
	191
	192
	193
	194
	195
	196
	197
	198
	199
	200
	201
	202
	203
	204
	205
	206
	207
	208
	209
	210
	211
	212
	213
	214
	215
	216
	217
	218
	219
	220
	221
	222
	223
	224
	225
	226
	227
	228
	229
	230
	231
	232
	233
	234
	235
	236
	237
	238
	239
	240
	241
	242
	243
	244
	245
	246
	247
	248
	249
	250
	251
	252
	253
	254
	255
	256
	257
	258
	259
	260
	261
	262
	263
	264
	265
	266
	267
	268
	269
	270
	271
	272
	273
	274
	275
	276
	277
	278
	279
	280
	281
	282
	283
	284
	285
	286
	287
	288
	289
	290
	291
	292
	293
	294
	295
	296
	297
	298
	299
	300
	301
	302
	303
	304
	305
	306
	307
	308
	309
	310
	311
	312
	313
	314
	315
	316
	317
	318
	319
	320
	321
	322
	323
	324
	325
	326
	327
	328
	329
	330
	331
	332
	333
	334
	335
	336
	337
	338
	339
	340
	341
	342
	343
	344
	345
	346
	347
	348
	349
	350
	351
	352
	353
	354
	355
	356
	357
	358
	359
	360
	361
	362
	363
	364
	365
	366
	367
	368
	369
	370
	371
	372
	373
	374
	375
	376
	377
	378
	379
	380
	381
	382
	383
	384
	385
	386
	387
	388
	389
	390
	391
	392
	393
	394
	395
	396
	397
	398
	399
	400
	401
	402
	403
	404
	405
	406
	407
	408
	409
	410
	411
	412
	413
	414
	415
	416
	417
	418
	419
	420
	421
	422
	423
	424
	425
	426
	427
	428
	429
	430
	431
	432
	433
	434
	435
	436
	437
	438
	439
	440
	441
	442
	443
	444
	445
	446
	447
	448
	449
	450
	451
	452
	453
	454
	455
	456
	457
	458
	459
	460
	461
	462
	463
	464
	465
	466
	467
	468
	469
	470
	471
	472
	473
	474
	475
	476
	477
	478
	479
	480
	481
	482
	483
	484
	485
	486
	487
	488
	489
	490
	491
	492
	493
	494
	495
	496
	497
	498
	499
	500
	501
	502
	503
	504
	505
	506
	507
	508
	509
	510
	511
	512
	513
	514
	515
	516
	517
	518
	519
	520
	521
	522
	523
	524
	525
	526
	527
	528
	529
	530
	531
	532
	533
	534
	535
	536
	537
	538
	539
	540
	541
	542
	543
	544
	545
	546
	547
	548
	549
	550
	551
	552
	553
	554
	555
	556
	557
	558
	559
	560
	561
	562
	563
	564
	565
	566
	567
	568
	569
	570
	571
	572
	573
	574
	575
	576
	577
	578
	579
	580
	581
	582
	583
	584
	585
	586
	587
	588
	589
	590
	591
	592
	593
	594
	595
	596
	597
	598
	599
	600
	601
	602
	603
	604
	605
	606
	607
	608
	609
	610
	611
	612
	613
	614
	615
	616
	617
	618
	619
	620
	621
	622
	623
	624
	625
	626
	627
	628
	629
	630
	631
	632
	633
	634
	635
	636
	637
	638
	639
	640
	641
	642
	643
	644
	645
	646
	647
	648
	649
	650
	651
	652
	653
	654
	655
	656
	657
	658
	659
	660
	661
	662
	663
	664
	665
	666
	667
	668
	669
	670
	671
	672
	673
	674
	675
	676
	677
	678
	679
	680
	681
	682
	683
	684
	685
	686
	687
	688
	689
	690
	691
	692
	693
	694
	695
	696
	697
	698
	699
	700
	701
	702
	703
	704
	705
	706
	707
	708
	709
	710
	711
	712
	713
	714
	715
	716
	717
	718
	719
	720
	721
	722
	723
	724
	725
	726
	727
	728
	729
	730

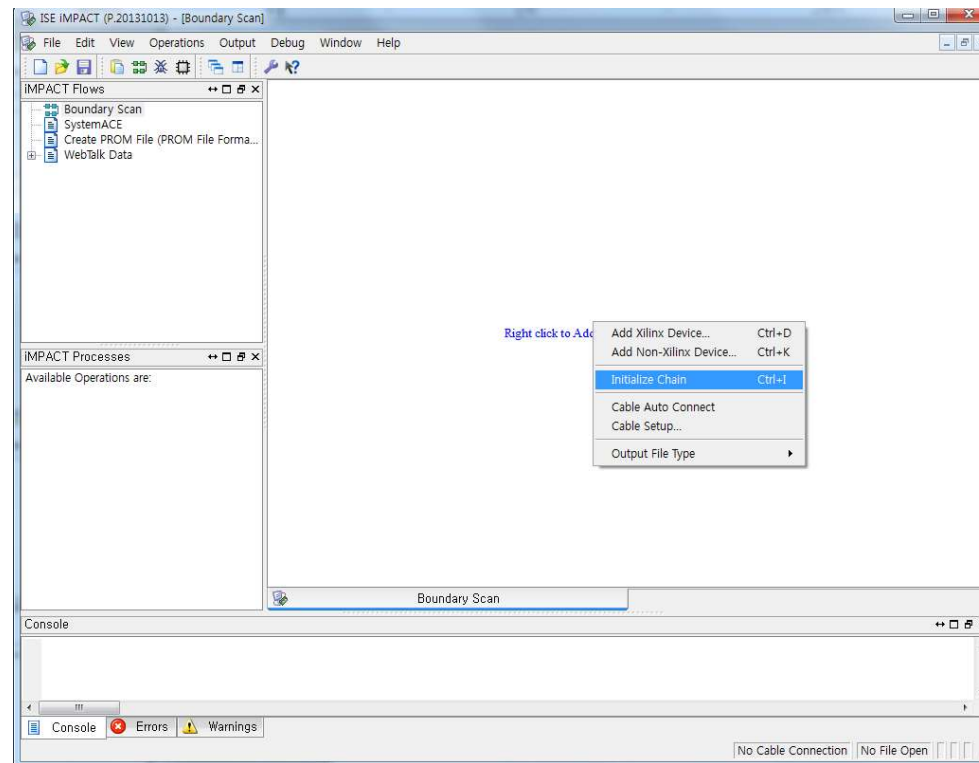
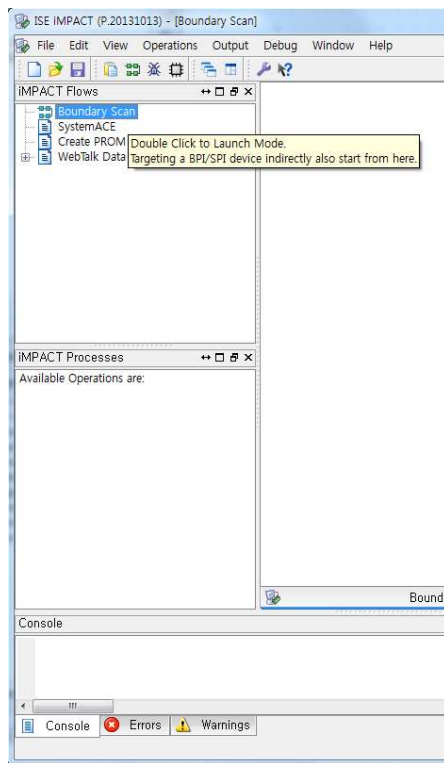
SNU Logic Design Board

- Assign I/O pins for the half adder



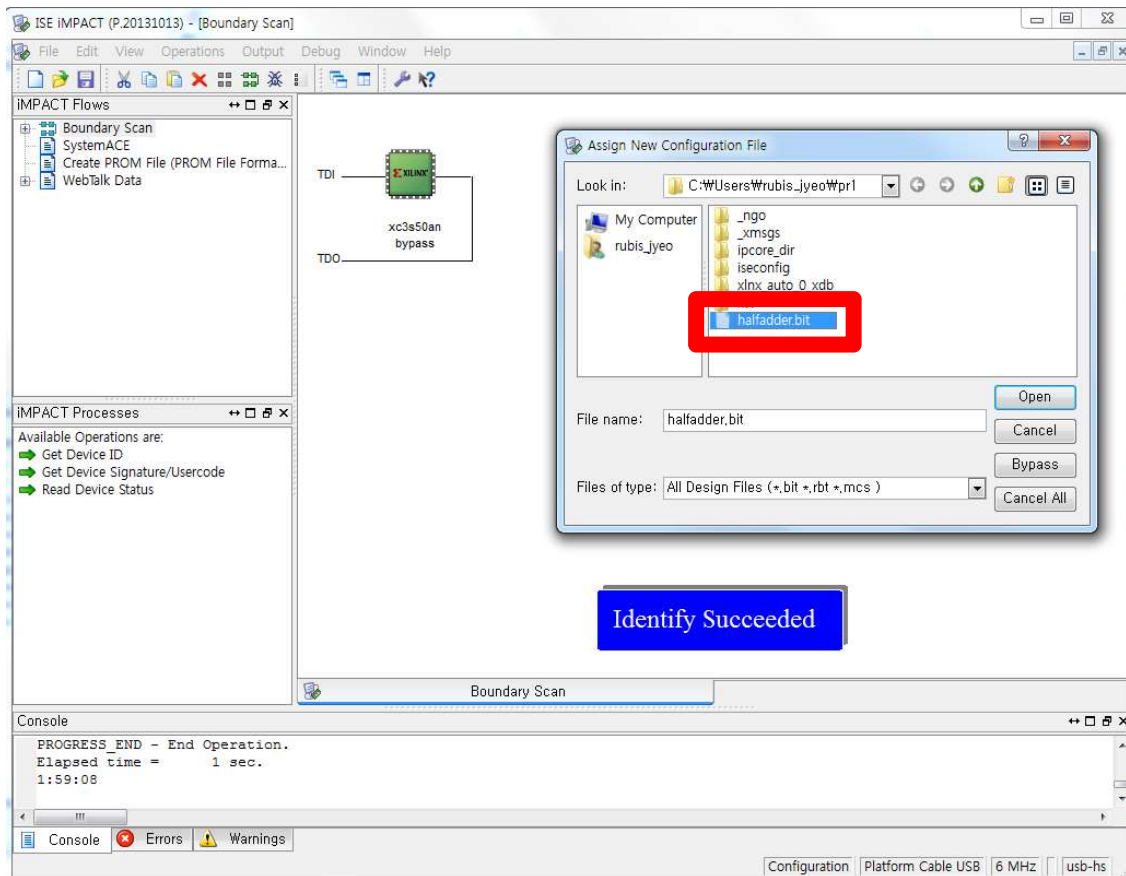
SNU Logic Design Board

- Boundary Scan > Initialize Chain



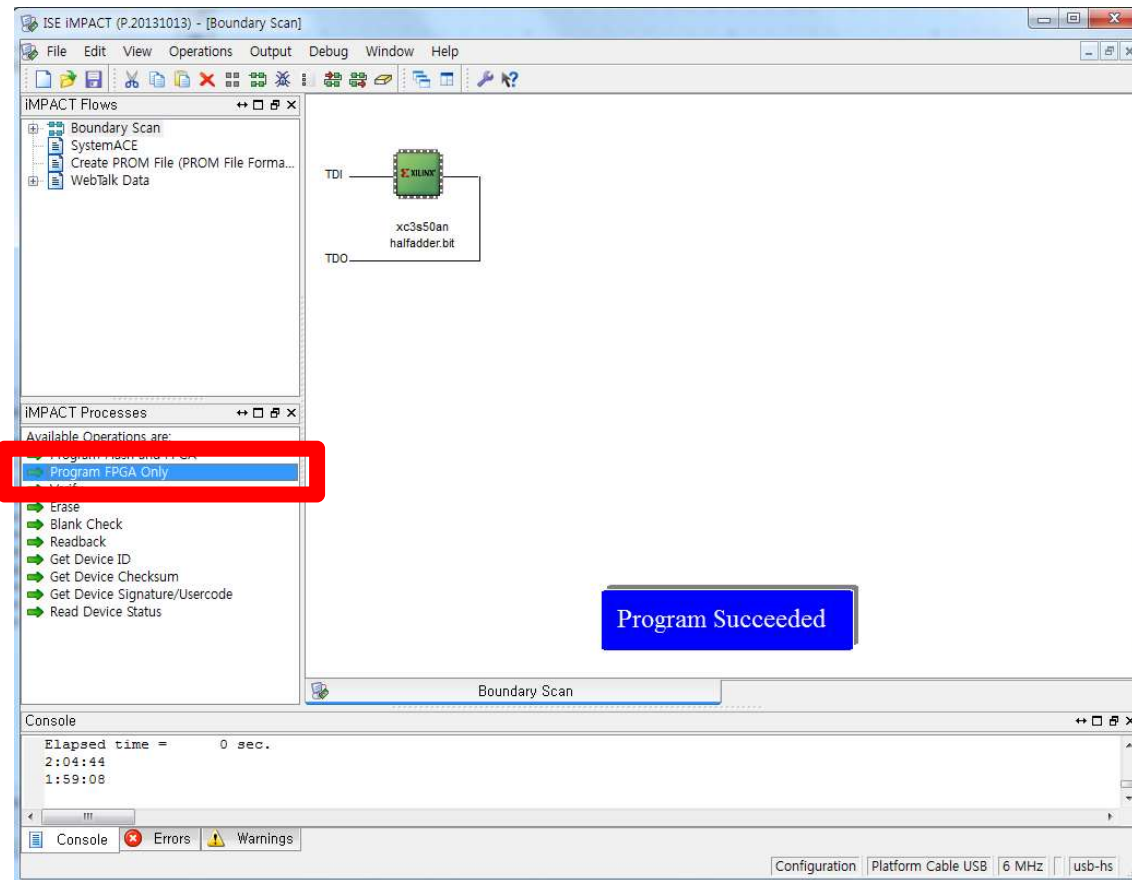
SNU Logic Design Board

- Choose the schematic source you've written.



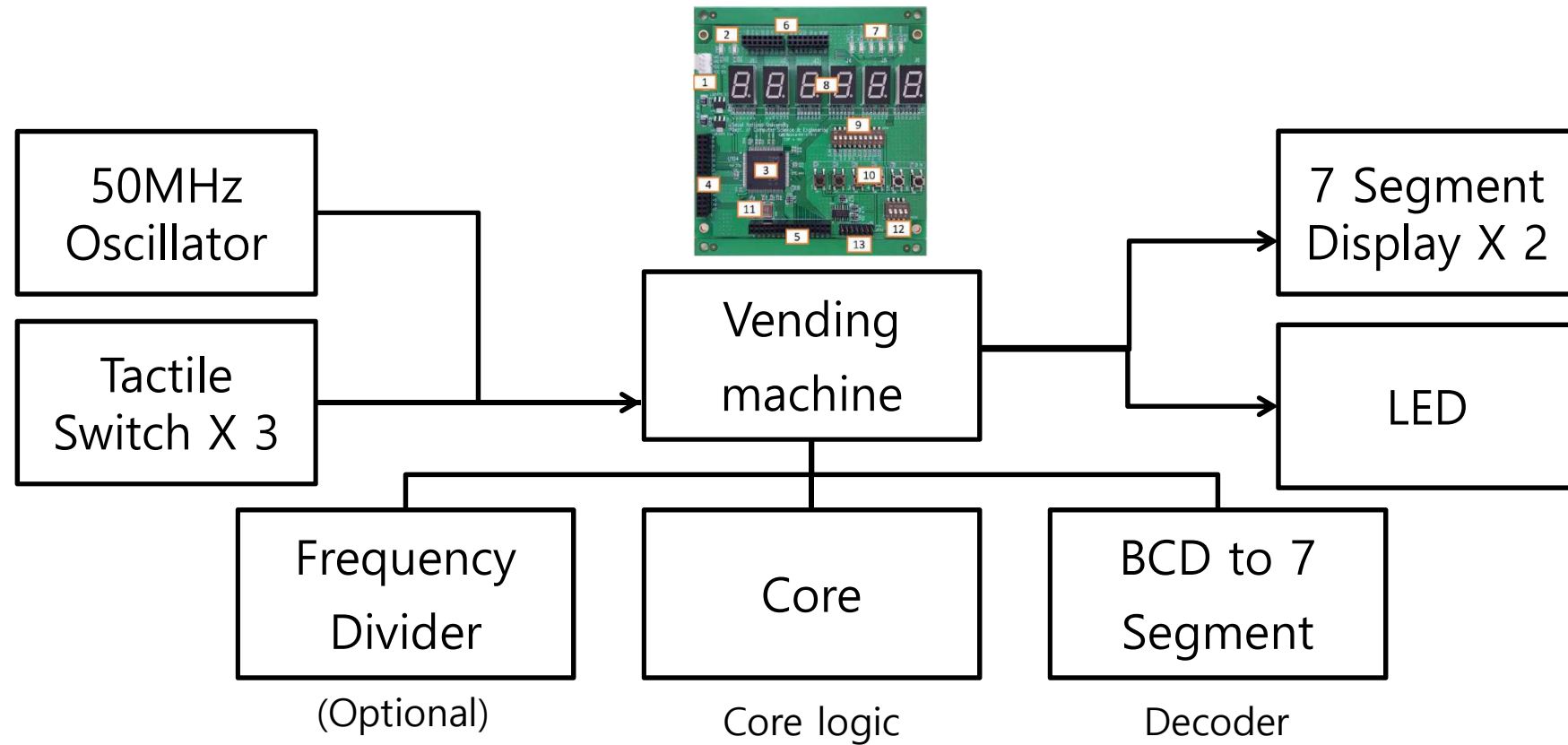
SNU Logic Design Board

- Program FPGA Only > Program Succeeded



LAB

Vending Machine Structure



LAB 1

1. Implement **core logic of vending machine** in Verilog. It accept nickels(5 cent) and dimes(10 cent). It has only single coin slot for nickels and dimes. The cost of a drink is 15 cents. It release item without change after 15 cents are deposited. Additionally, your implementation should handle external asynchronous reset signal, it should set the deposit back to 0 when the signal is HIGH. If the deposit is greater than 15 cents, consider it as 15 cents.

LAB 2

2. Implement the **Vending machine** in Verilog and upload your implementation to the Logic Design Board. It accept nickels(5 cent) and dimes(10 cent). It has only single coin slot for nickels and dimes. The cost of a drink is 15 cents. It release item without change after 15 cents are deposited. Additionally, your implementation should handle external asynchronous reset signal, it should set the deposit back to 0 when the signal is HIGH. If the deposit is greater than 15 cents, consider it as 15 cents. Use 50MHz oscillator as clock input, and a tactile switch as reset button and coin inputs. For outputs, use two 7-segment displays as your deposit status, and led as release status.

Report

- **Core logic of vending machine**

- 1. Result of simulation

- All possible case of coin inputs
 - Deposit and release status should be 0 when reset = 1

- **Vending machine**

- 1. All Verilog source code

- should include code for module, submodule (i.e., Vending machine, core logic, BCD-7 seg decoder)
 - only module code, **no test bench code**

- 2. Result of implementation

- Pictures of all possible states that you design

- **Discussion**

(Optional) Study and compare Moore and Mealy machine

Sample Code (Behavioral Description)

```
`timescale 1ns / 1ps

module freq_divider(
    input clr,
    input clk,
    output reg clkout
);
    reg[31:0] cnt;
    always@ (posedge clk) begin
        if(clr) begin
            cnt <= 32'd0;
            clkout <= 1'b0;
        end
        else if (cnt == 32'd25000000) begin
            cnt <= 32'd0;
            clkout <= ~clkout;
        end
        else begin
            cnt <= cnt + 1;
        end
    end
end

endmodule
```

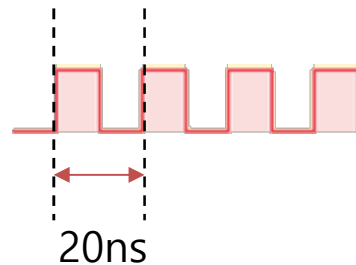
Frequency Divider

```
1 `timescale 1ns / 1ps
2
3 module bcd_to_7(
4     input [3:0] bcd,
5     output reg [6:0] seg
6 );
7
8     always@(bcd) begin
9         case(bcd)
10             4'd0: seg <= 7'b0111111;
11             4'd1: seg <= 7'b0000110;
12             4'd2: seg <= 7'b1011011;
13             4'd3: seg <= 7'b1001111;
14             4'd4: seg <= 7'b1100110;
15             4'd5: seg <= 7'b1101101;
16             4'd6: seg <= 7'b1111101;
17             4'd7: seg <= 7'b0000111;
18             4'd8: seg <= 7'b1111111;
19             4'd9: seg <= 7'b1101111;
20         endcase
21     end
22
23 endmodule
```

BCD to 7-segment Decoder

Homework

- Simulating 50MHz clock in test bench
 - 50MHz : period is 20ns
(So we should flip clock every 10ns)



```
`timescale 1ns / 1ps

module counter_sim;

    // Inputs
    reg clk;
    reg reset;

    // Outputs
    wire [6:0] cnt;

    // Instantiate the Unit Under Test (UUT)
    counter uut (
        .clk(clk),
        .reset(reset),
        .cnt(cnt)
    );

    always #10 clk=~clk;

    initial begin
        // Initialize Inputs
        clk = 0;
        reset = 1;
    end

endmodule
```

Report

- Write a report
 - Either in Korean or in English
 - **Your report should include**
 - Discussion
 - Homework
 - # of pages doesn't matter
 - Documents should be submitted as PDF file(**less than 25Mb**)
 - **Attach source code and waveform screenshot**
 - Due :
 - Class 001 - May, 23th (Before class begin at 7:00pm)
 - Class 002 - May, 24th (Before class begin at 7:00pm)
 - Class 003 - May, 25th (Before class begin at 7:00pm)