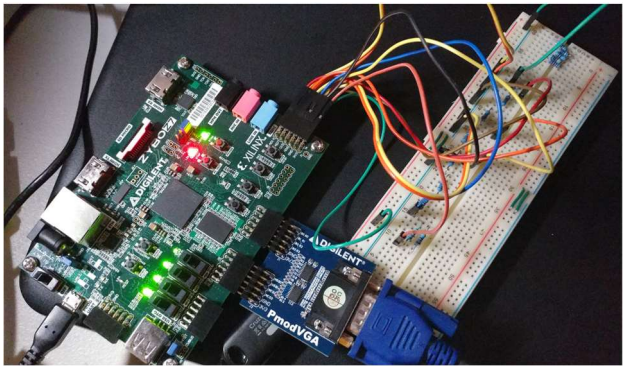
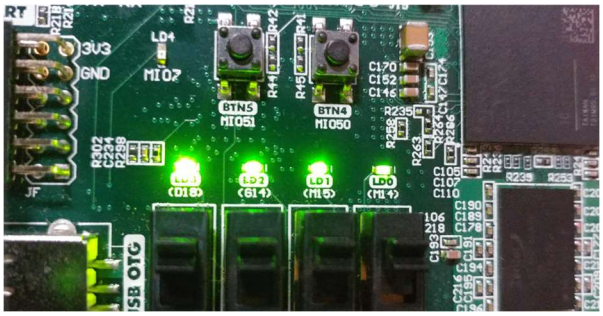
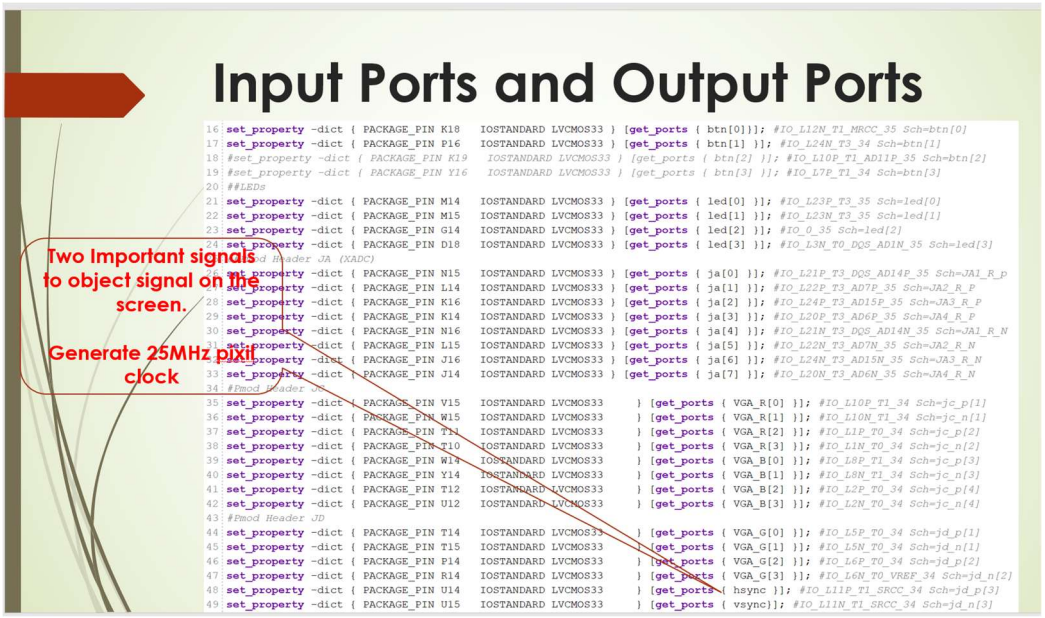
Julie Kim (Engineering Portfolio)

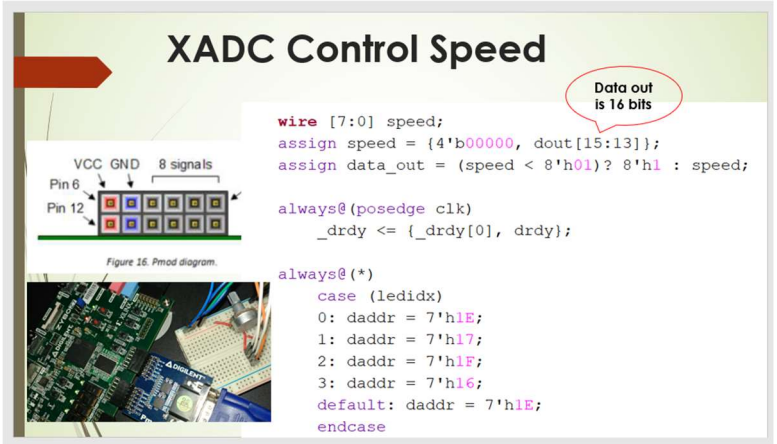
Long Beach Ca 90813 | 562-607-1465 | chealy_ljh@yahoo.co.uk

- RTL Analog to Digital Converter Custom IP (*Spring 2021 – FPGA Based SW&HW Co-Desgn)*
  https://www.youtube.com/watch?v=VVtr0a4FmV0&list=PLWB9IGreRgxyr_xbJwedF16atCr6rWCUh&index=2





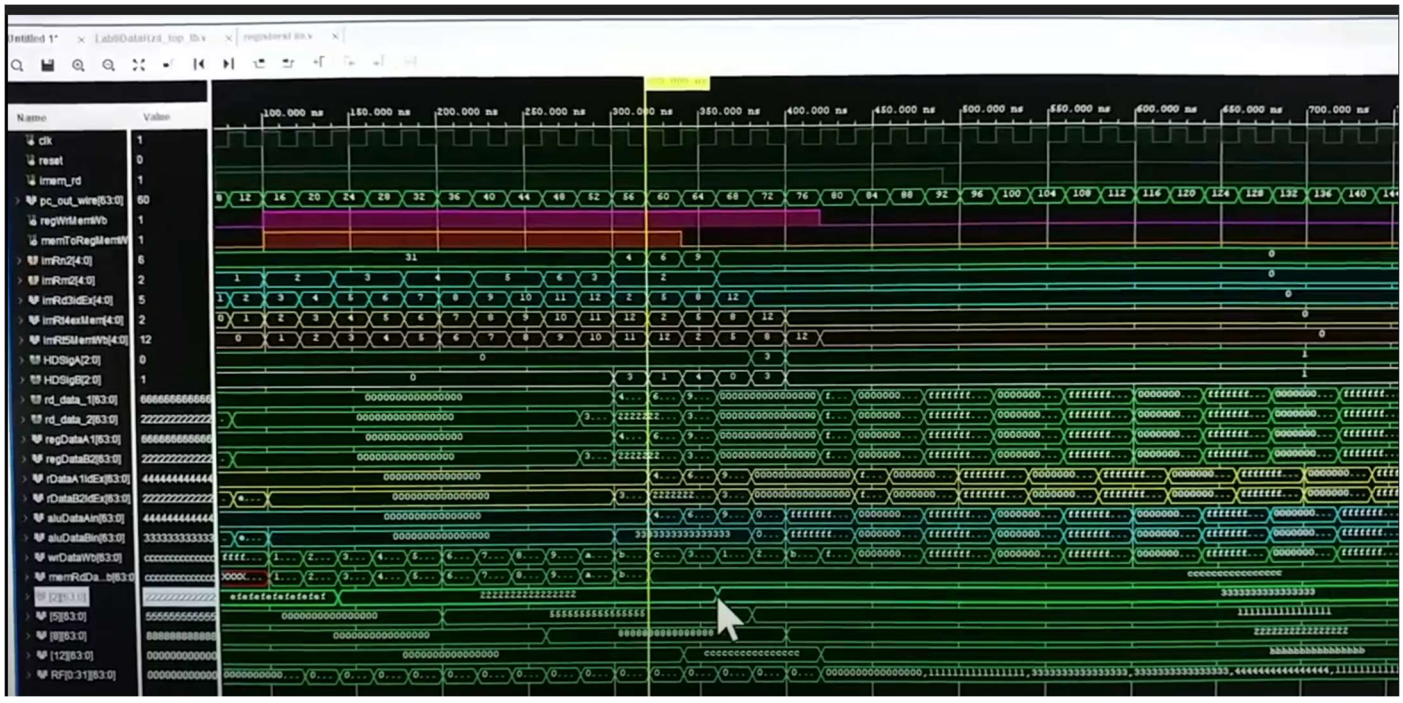




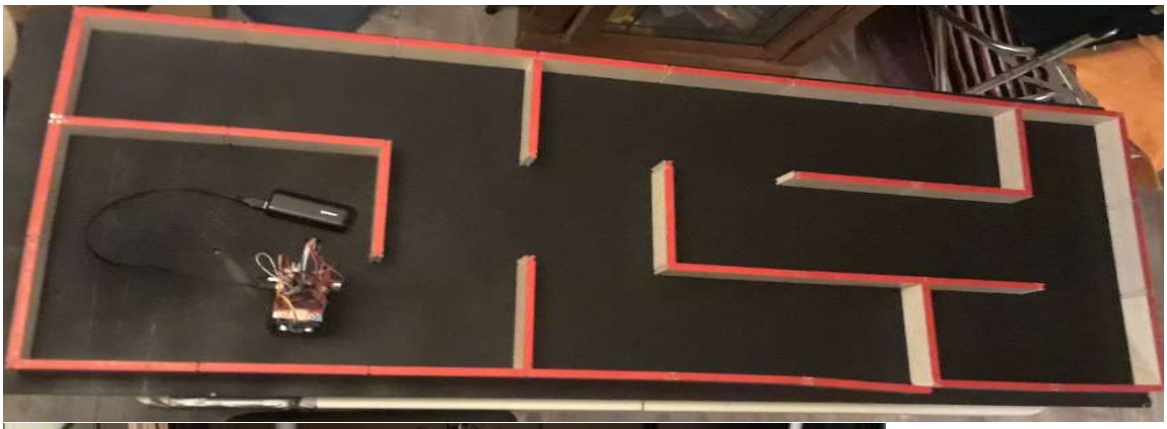
- RISC 32-bit 5 stages Pipelined with Forwarding Unit and Hazard Detector (*Fall 2020 – Adv Computer Architecture I*)
  https://drive.google.com/file/d/1UGjc_aACZLHmnbcP7u7_3vsWSSOOd89l/view?usp=sharing

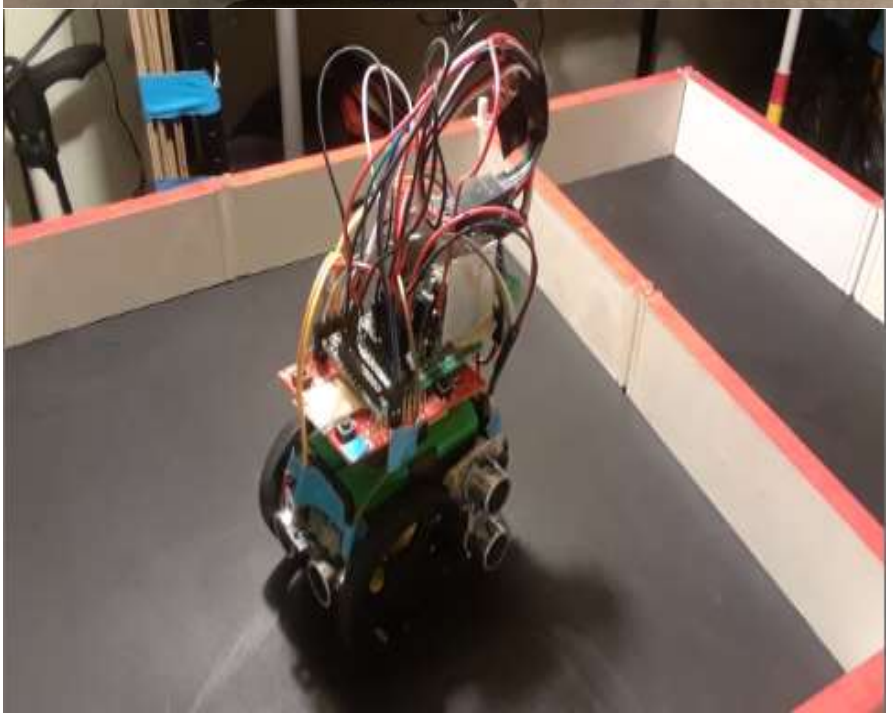
```
module registersFile (clk, reset, wr_en, wr_addr, wr_data,
                      rd_addr_1, rd_addr_2, rd_data_1,
                      rd_data_2);
    input clk, reset, wr_en;
    input [4:0] rd_addr_1, rd_addr_2, wr_addr;
    input [63:0] wr_data;
    output wire [63:0] rd_data_1, rd_data_2;

    reg [63:0] RF[0:31];
    //reg [63:0] dtA1, dtB2;
    integer i;

    always @(posedge clk, posedge reset)
```

```
157                InstructionMem1.RF[56 + 3] = 8'b110_01000;
158
159                            //x2
160                //OR X12, X9, X2  ;  Rn, Rm    ; data1, data2
161                //32'b10101010000_00010_000000_01001_01100;
162                //32'b10101010 000_00010_ 000000_01 001_01100;
163                InstructionMem1.RF[60 + 0] = 8'b10101010;
164                InstructionMem1.RF[60 + 1] = 8'b000_00010;  //Rm==2
165              // InstructionMem1.RF[60 + 1] = 8'b000_01010;  //Rm==10
166                InstructionMem1.RF[60 + 2] = 8'b000000_01;
167                InstructionMem1.RF[60 + 3] = 8'b001_01100;
168
169              @(negedge clk)
170              reset = 1'b1;
171              @(negedge clk)
172              reset = 0;
173              imem_rd = 1'b1;
```

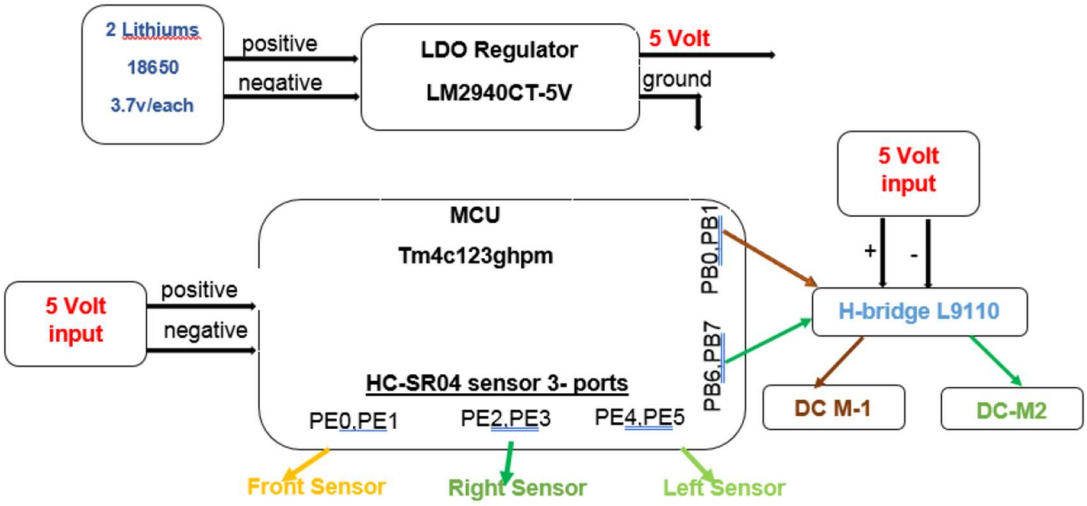


- Real-Time Autonomous Robot (*Fall 2018 - Computer Engr Snr Proj II*)
  https://www.youtube.com/watch?v=9__2CTwlCjk&list=PLWB9IGreRgxzoJ4VC523rkH4-hUHCCX6L&index=1

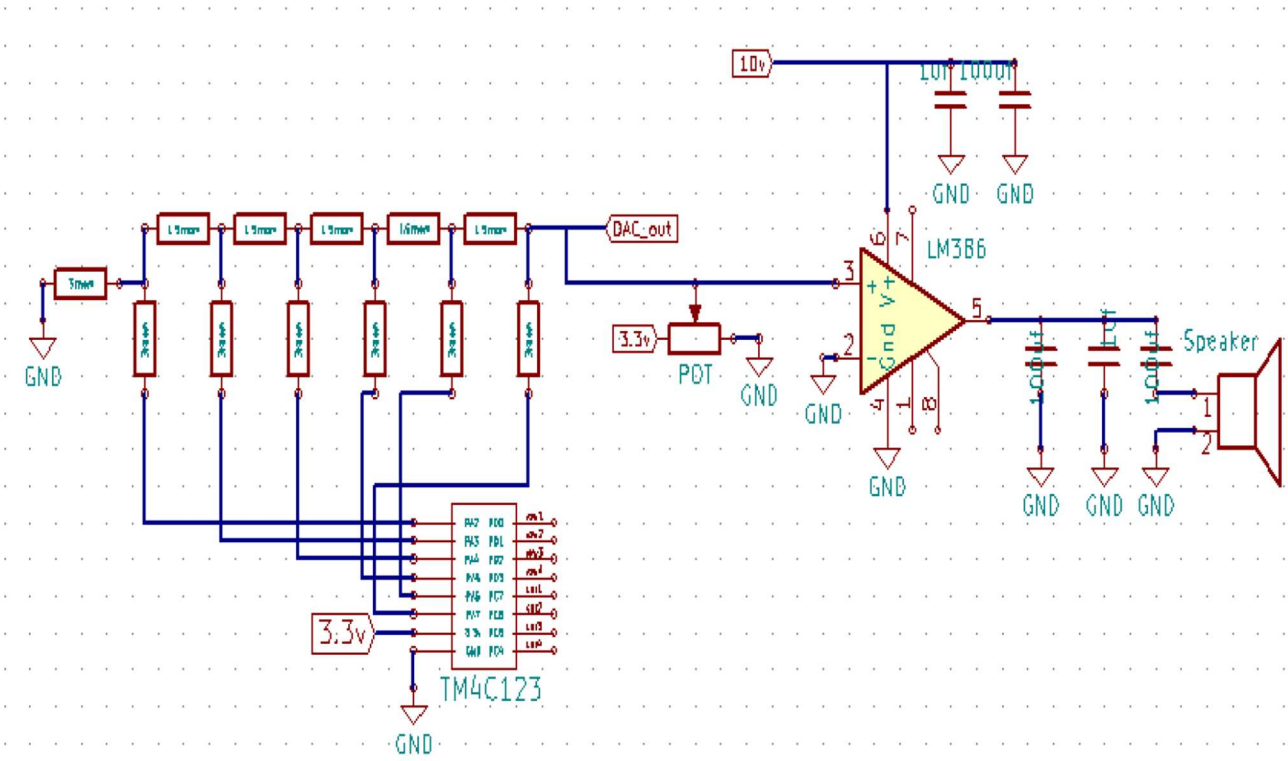




Overall System Functional Block Diagram:



- R2R Digital to Analog Convertor (Spring 2018 – Microprocessor+Ctr III)



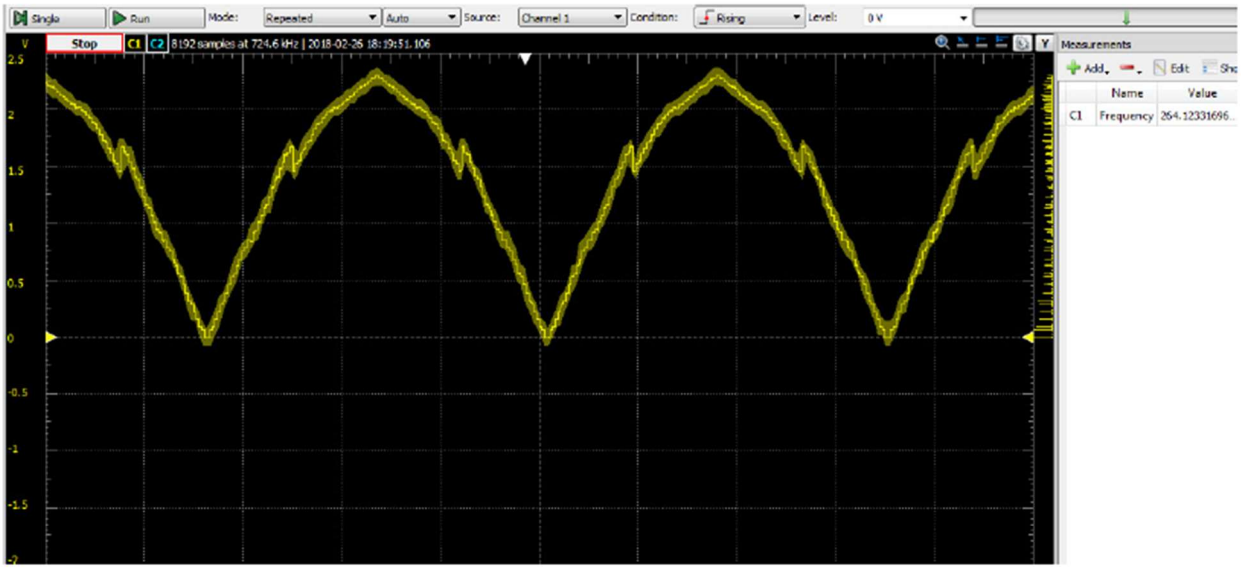
Truth Table:

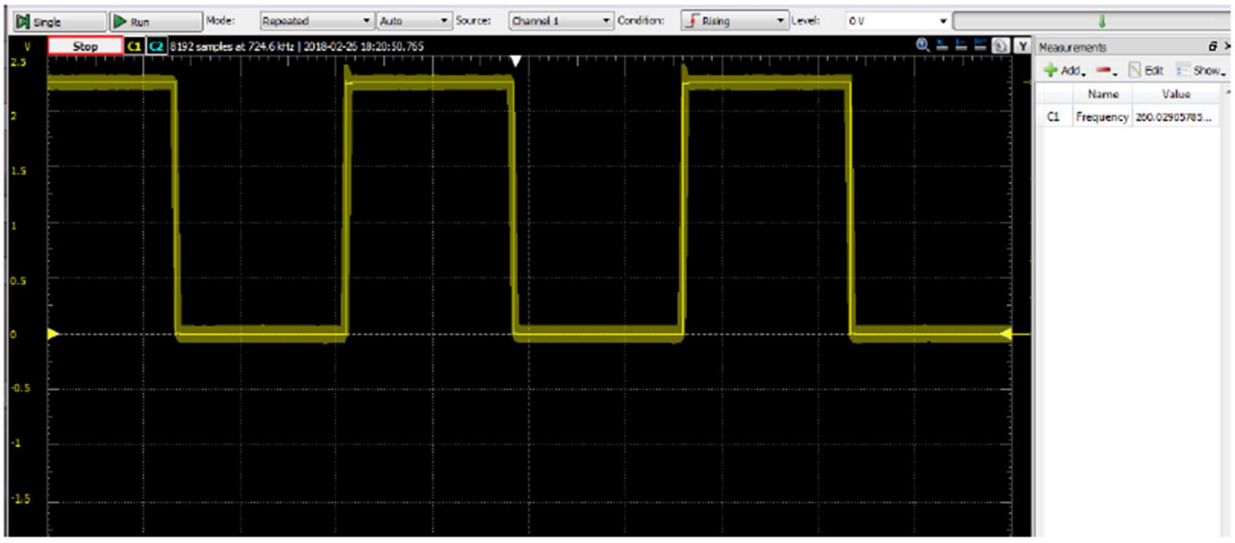
Calibrating Voltage Value to Binary value

	Decimal	Binary Value						V_out
		D5	D4	D3	D2	D1	D0	Voultage
3.3	0	0	0	0	0	0	0	0
3.3	1	0	0	0	0	0	1	0.0515625
3.3	2	0	0	0	0	1	0	0.103125
3.3	3	0	0	0	0	1	1	0.1546875
3.3	4	0	0	0	1	0	0	0.20625
3.3	5	0	0	0	1	0	1	0.2578125
3.3	6	0	0	0	1	1	0	0.309375
3.3	7	0	0	0	1	1	1	0.3609375
3.3	8	0	0	1	0	0	0	0.4125
3.3	9	0	0	1	0	0	1	0.4640625
3.3	10	0	0	1	0	1	0	0.515625
3.3	11	0	0	1	0	1	1	0.5671875
3.3	12	0	0	1	1	0	0	0.61875
3.3	13	0	0	1	1	0	1	0.6703125
3.3	14	0	0	1	1	1	0	0.721875
3.3	15	0	0	1	1	1	1	0.7734375
3.3	16	0	1	0	0	0	0	0.825
3.3	17	0	1	0	0	0	1	0.8765625
3.3	18	0	1	0	0	1	0	0.928125
3.3	19	0	1	0	0	1	1	0.9796875
3.3	20	0	1	0	1	0	0	1.03125
3.3	21	0	1	0	1	0	1	1.0828125
3.3	22	0	1	0	1	1	0	1.134375
3.3	23	0	1	0	1	1	1	1.1859375
3.3	24	0	1	1	0	0	0	1.2375

Logic Analyzer Simulation of R2R Digital to Analog Convertor circuit:

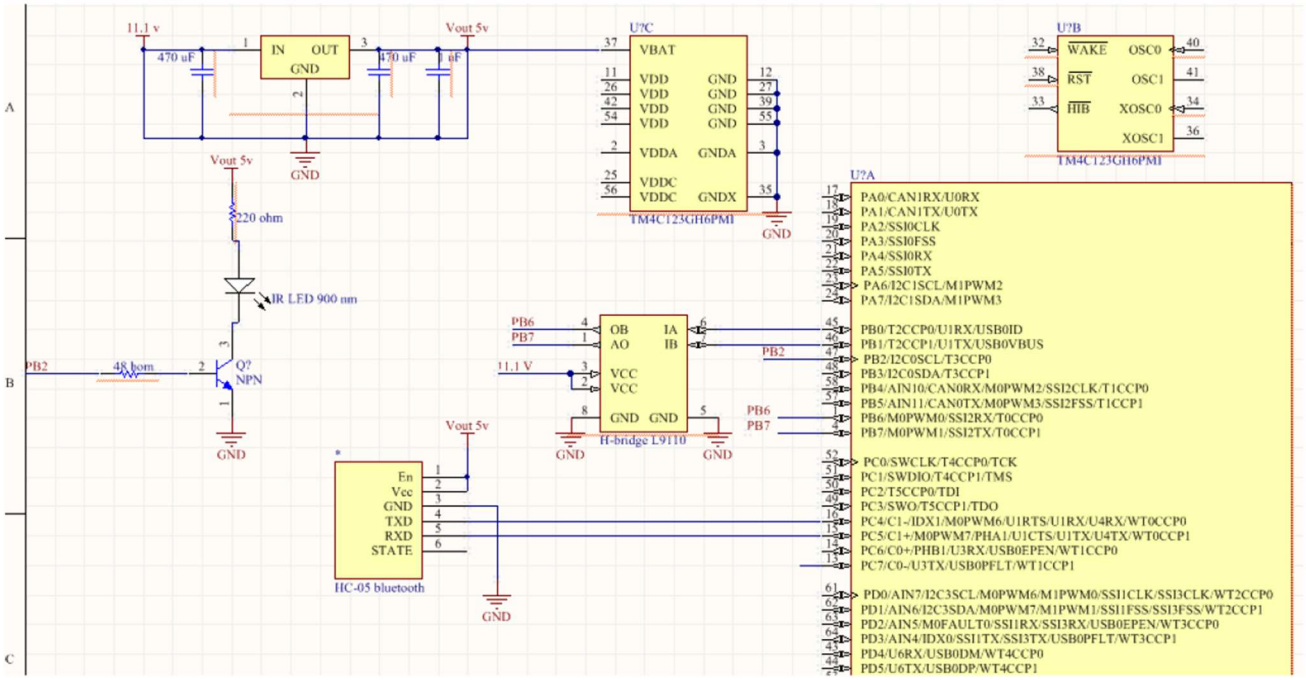
Figures:

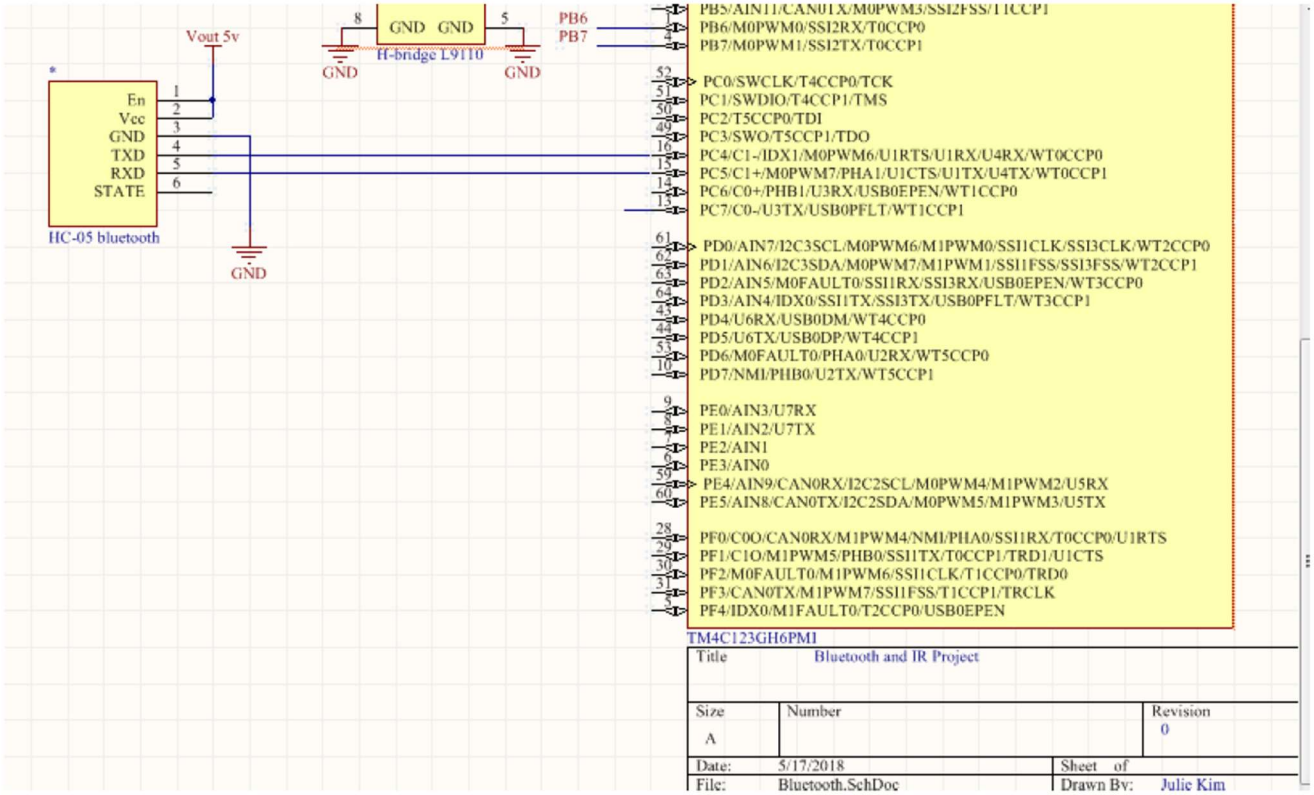




- Bluetooth Controlled Robot (*Spring 2018 – Microprocessor+Ctr III*)

Schematics:

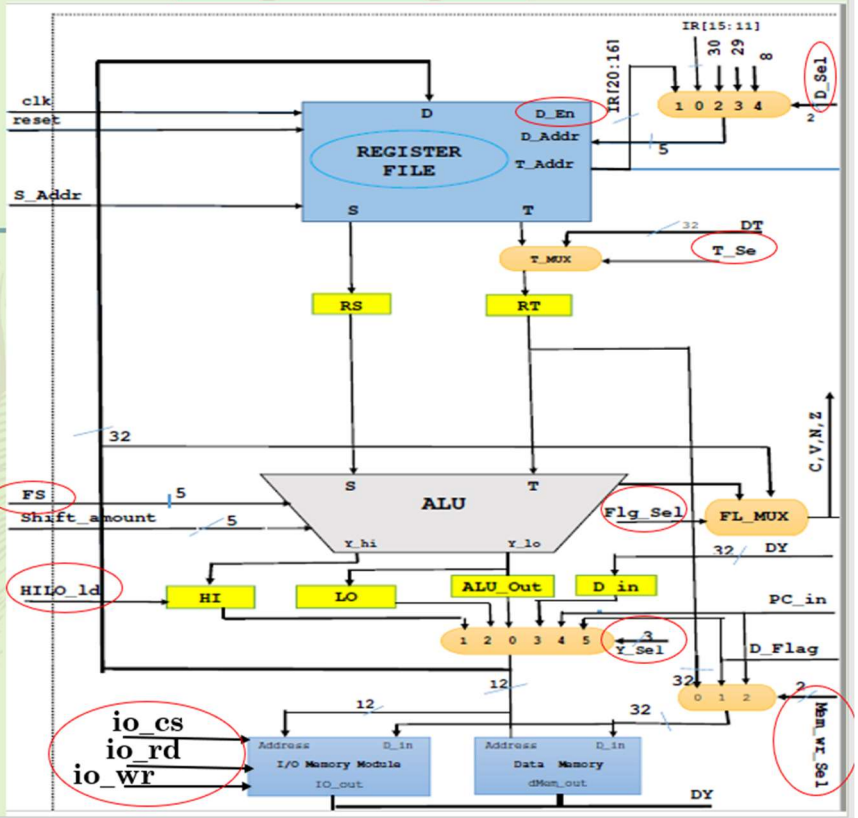




- MIPS Non-Pipelined 32-bit CPU with Barrel Shifter (*Computer Architecture – Fall 2017*)
  https://www.youtube.com/watch?app=desktop&v=e0dxrGOGR4c&t=483s

Integer Datapath:

- Memory Address, 12bit
- Memory Data in, 32bit
- D_in: IDP, DY, PC, D_Flag
- Memory Output: DY

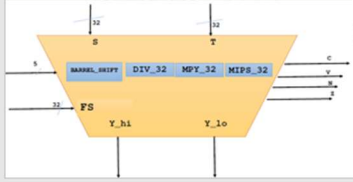


Barrel Shift Left Result Verification:

```
/*****************************************
*BARREL SHIFT LEFT LOGICAL 32
*****************************************/
     {5'h0C, 5'd1}:   {c, y_lo} = {TData[31], {TData[30:0], 1'b0}};        //SLL_1
     {5'h0C, 5'd2}:   {c, y_lo} = {TData[30], {TData[29:0], 2'b0}};        //SLL_2
     {5'h0C, 5'd3}:   {c, y_lo} = {TData[29], {TData[28:0], 3'b0}};        //SLL_3
     {5'h0C, 5'd4}:   {c, y_lo} = {TData[28], {TData[27:0], 4'b0}};        //SLL_4
     {5'h0C, 5'd5}:   {c, y_lo} = {TData[27], {TData[26:0], 5'b0}};        //SLL_5
     {5'h0C, 5'd6}:   {c, y_lo} = {TData[26], {TData[25:0], 6'b0}};        //SLL_6
     {5'h0C, 5'd7}:   {c, y_lo} = {TData[25], {TData[24:0], 7'b0}};        //SLL_7
     {5'h0C, 5'd8}:   {c, y_lo} = {TData[24], {TData[23:0], 8'b0}};        //SLL_8
     {5'h0C, 5'd9}:   {c, y_lo} = {TData[23], {TData[22:0], 9'b0}};        //SLL_9
```

Opcode					Function
0x00	$rs	$rt	$rd	shmt	0x00

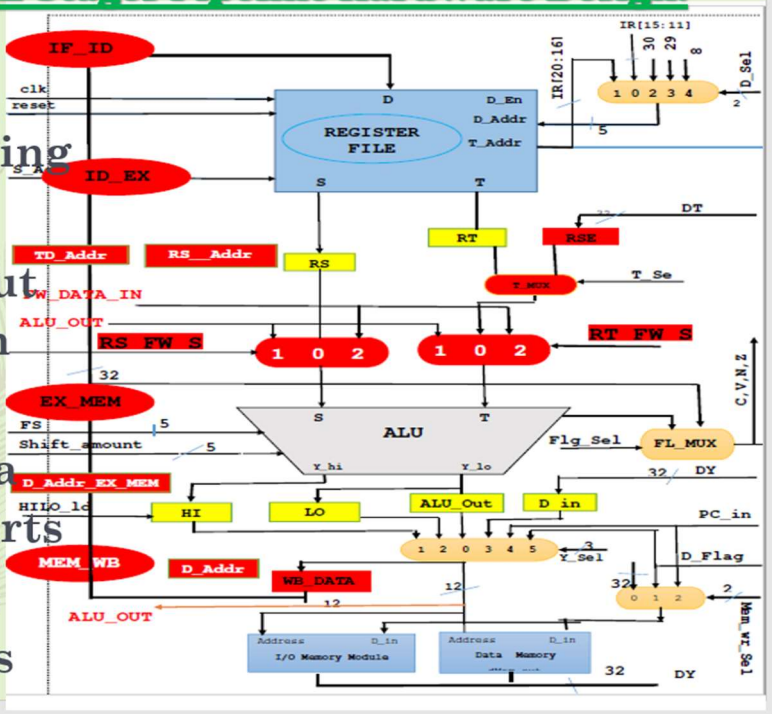
ALU Internal Design

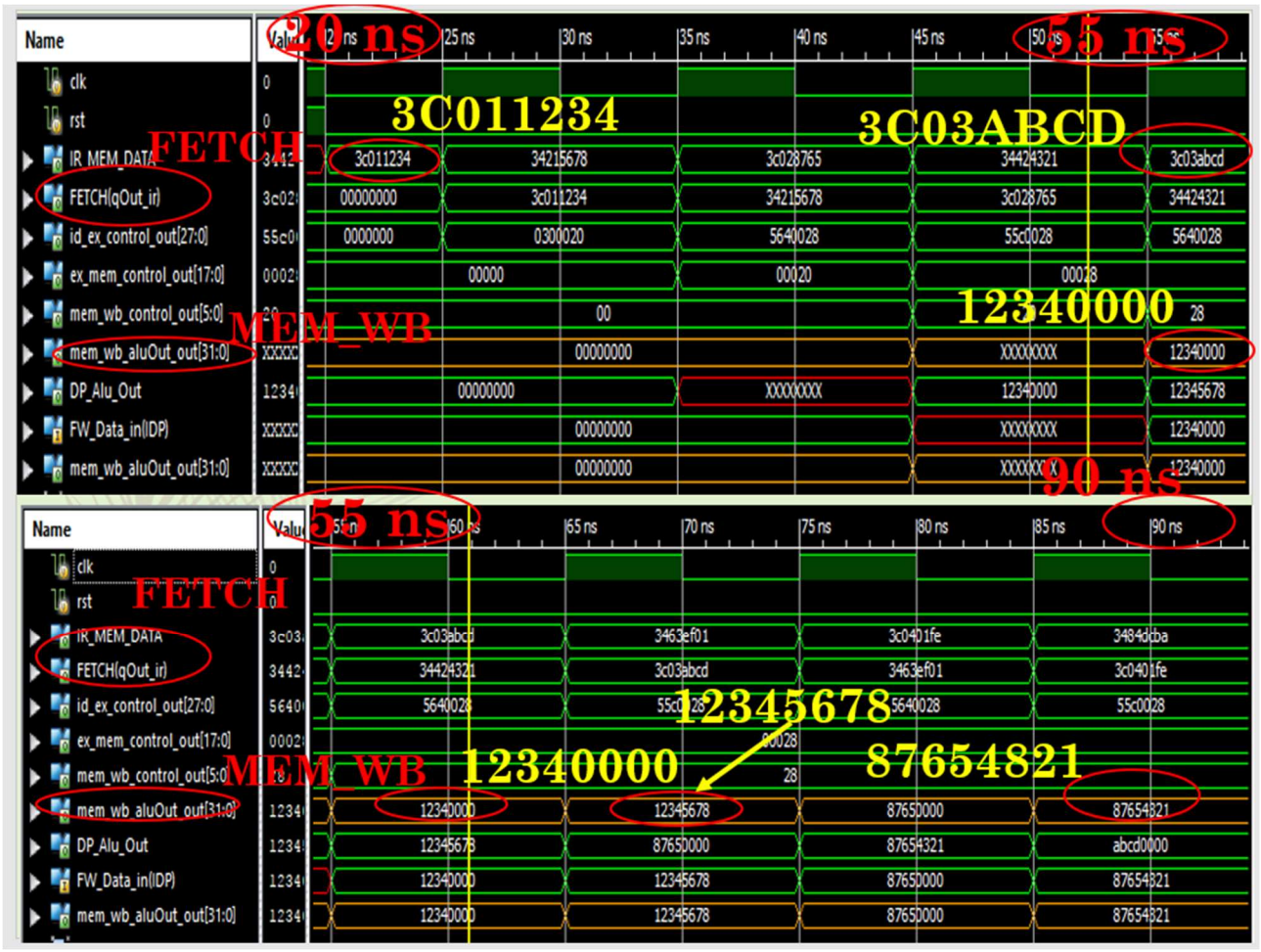


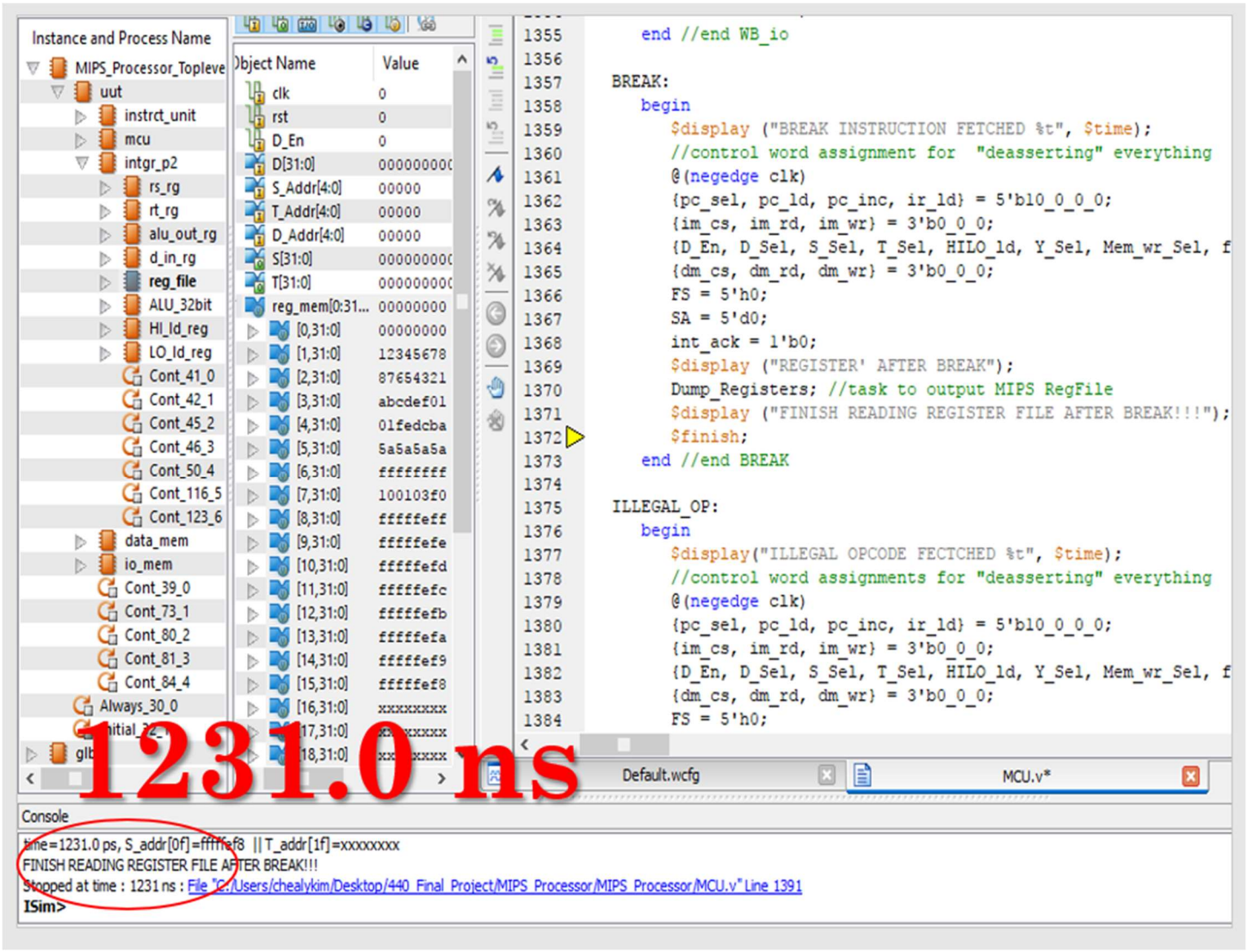

```
**********************************************************************
C E C B 4 4 0 A_L_U T e s t b e n c h R e s u l t s
----------------------------------------------------------------------
SHIFT LEFT LOGICAL:
Finished circuit initialization process.
SLL 04: T=800ffff0 Ylo=00ffff00 c=0 v=0 n=0 z=0
SLL 08: T=800ffff0 Ylo=0ffff000 c=0 v=0 n=0 z=0
SLL 16: T=800ffff0 Ylo=fff00000 c=1 v=0 n=1 z=0
SLL 20: T=800ffff0 Ylo=ff000000 c=1 v=0 n=1 z=0
```

FIVE Stages Pipeline Hardware Design:

- Forwarding Unit
- Two Input Selection ports
- Two Data Input Ports
- Six more Registers







End of Portfolio. Thank you for taking the time to review.

Sincerely,

Julie Kim

Julie Kim