ICEEP
Iran Center of ESL Education and Promotion

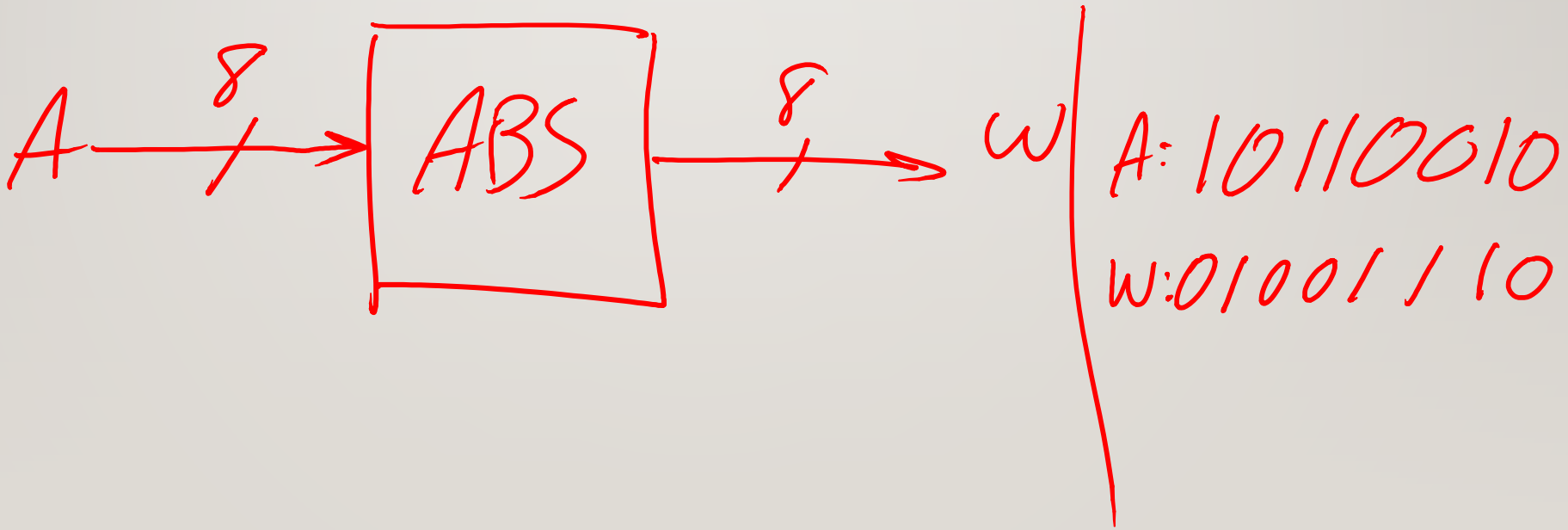
Digital Logic Design

Lecture 1

Dr. Navabi

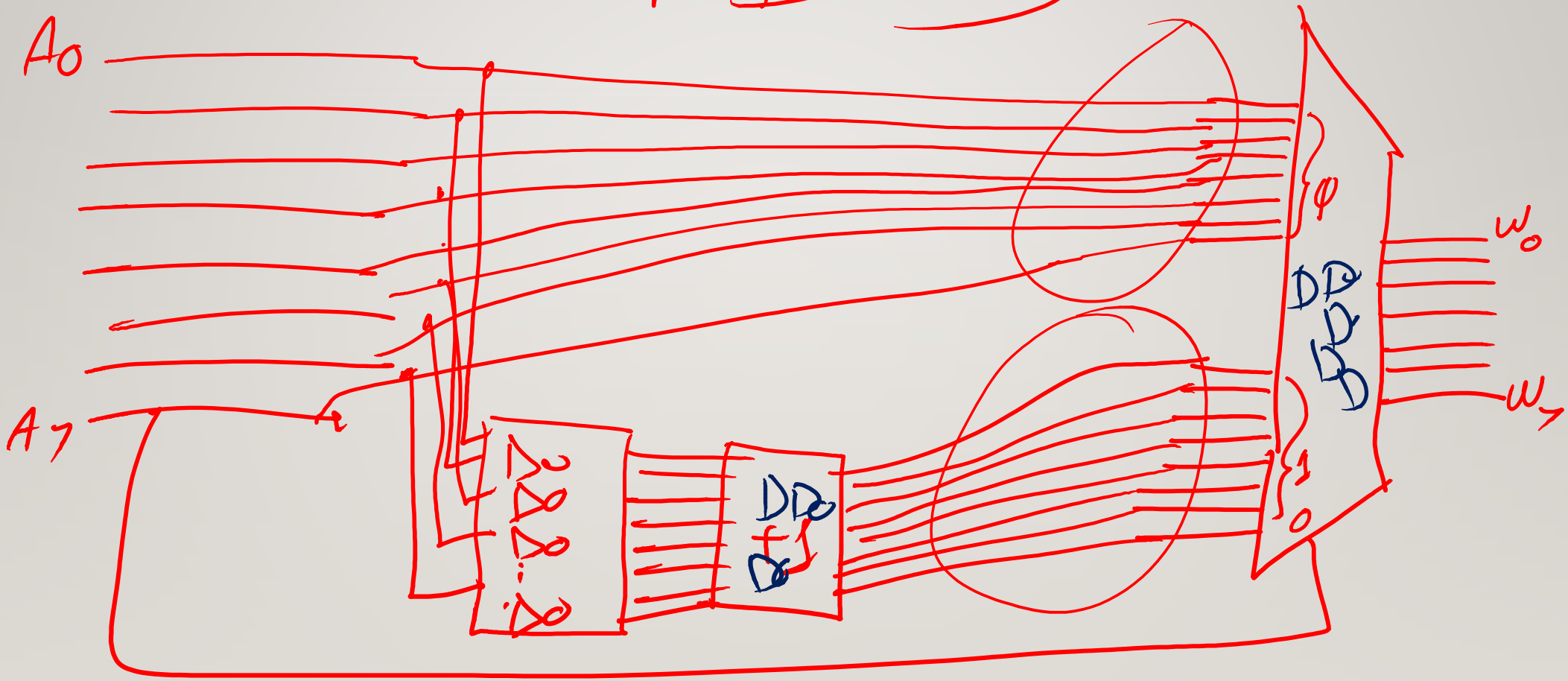
Lecture 1 RTL

Register Transfer Level

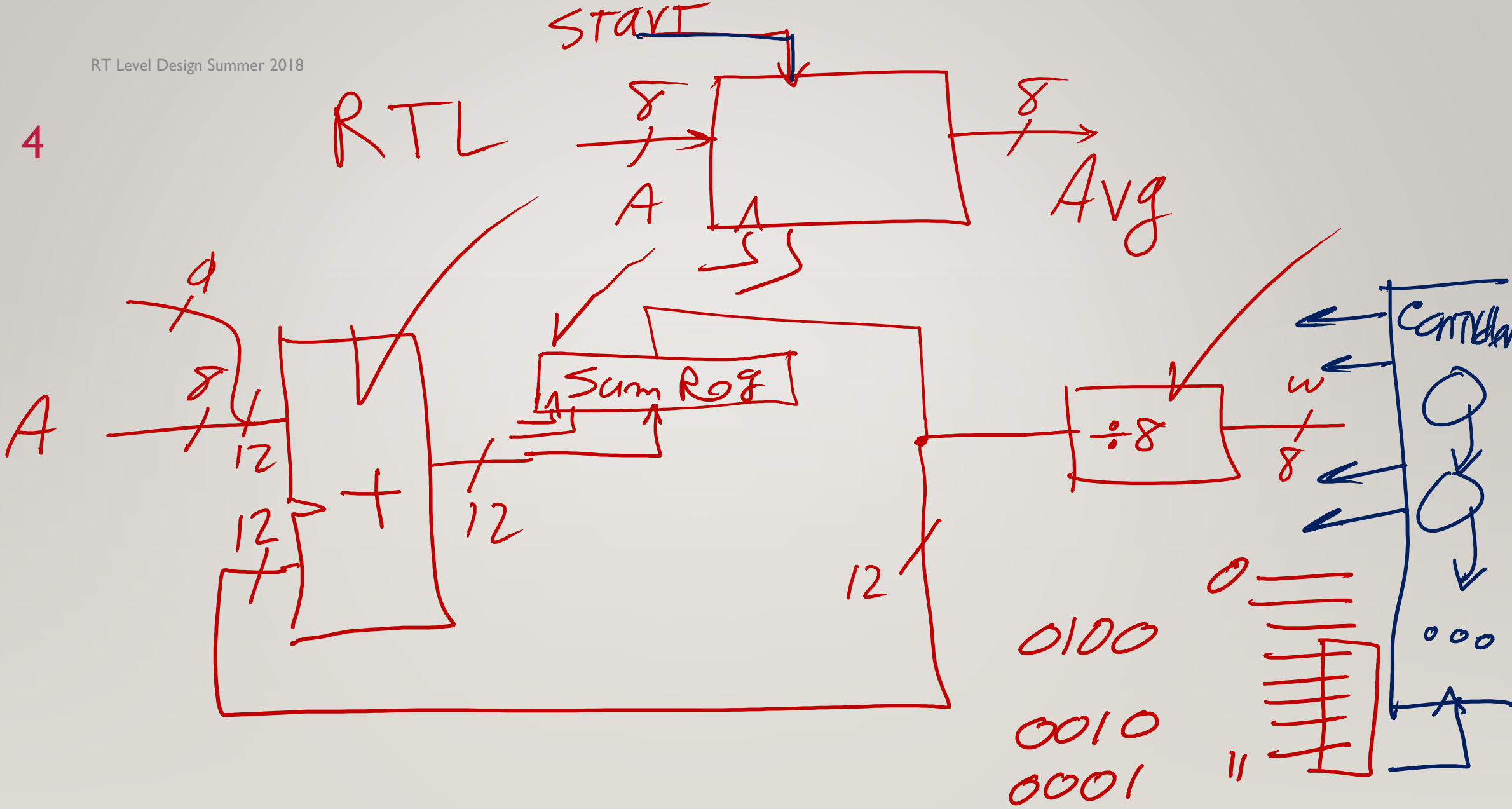


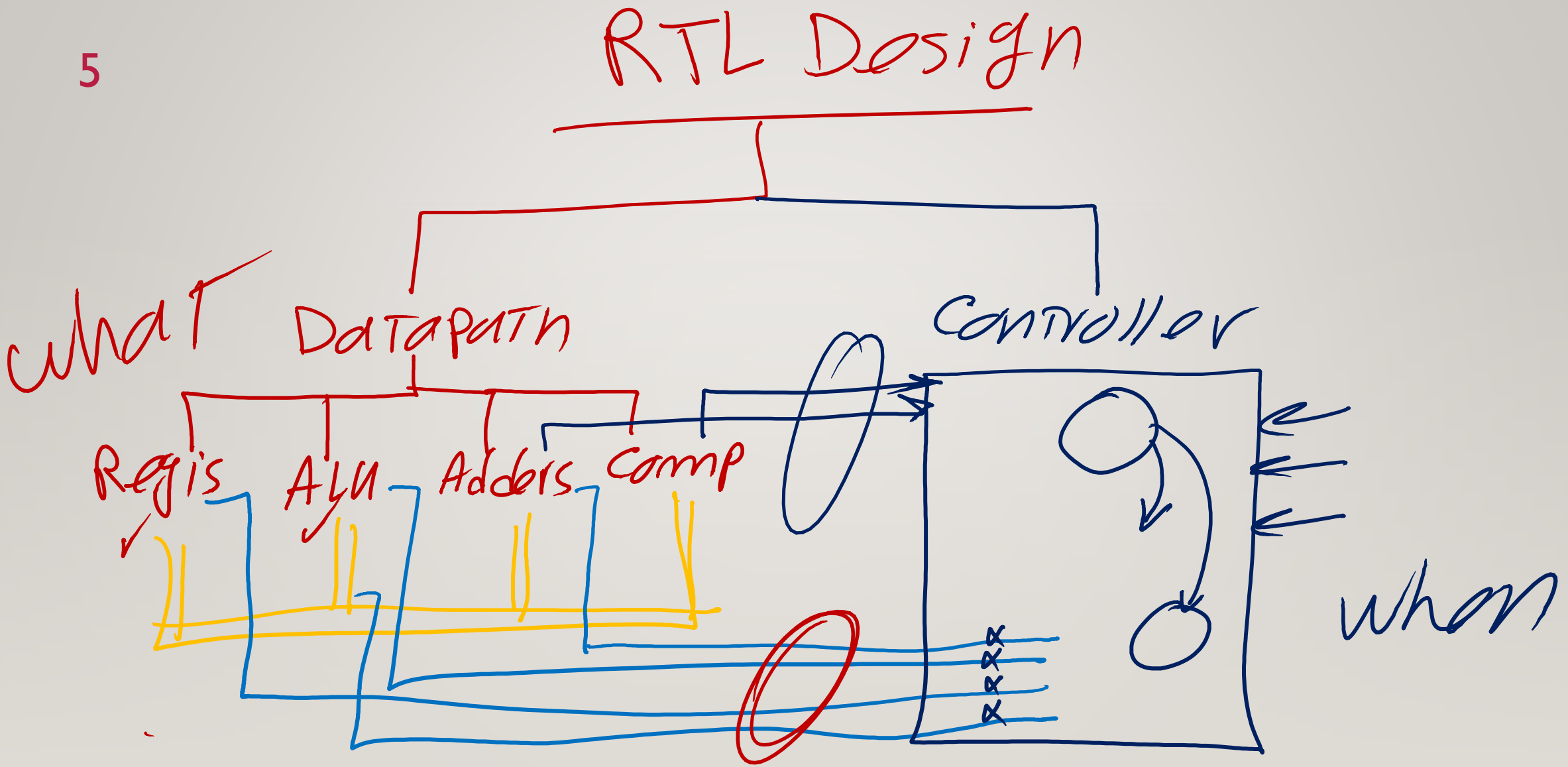
3

ABS

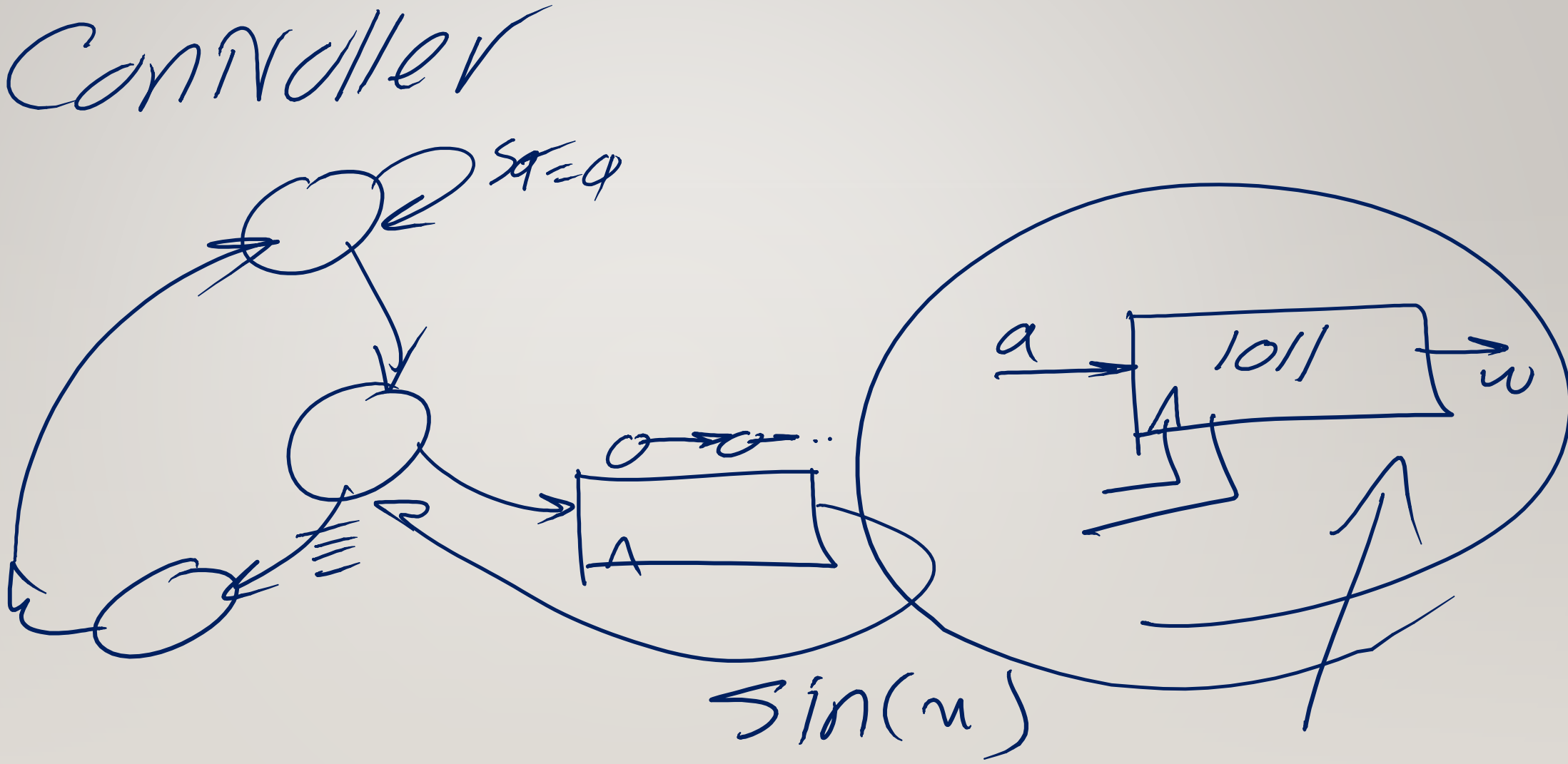


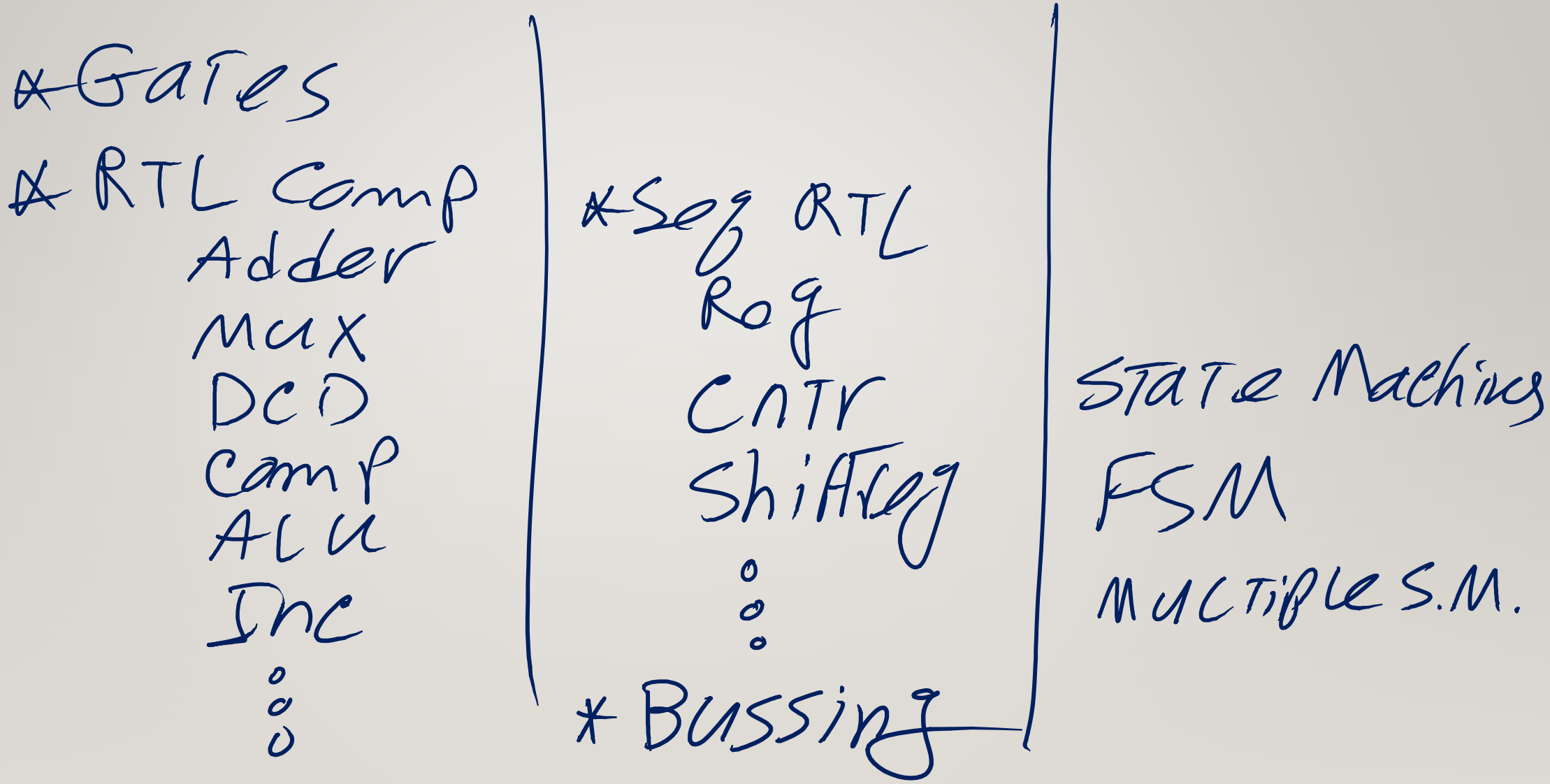
4


START
RTL
8
8
A
A
Avg
9
8
A
12
12
12
12
Sum Reg
÷8
8
w
0
0100
0010
0001
11
000


RTL Design
what
Datapath
Controller
Regis
Alu
Adders
comp
when

6




*Gates
*RTL Comp
Adder
Mux
DCD
Comp
ALU
Inc
*Seq RTL
Reg
Cntr
Shiftreg
*Bussing
State Machine
FSM
Multiple S.M.

8

* RTL comp.
* RTL Method.
* Inter_RTL comm.
* wrappers

9 Gates


```
module myand(input a, b, output w);
    and #(3,5) u1 (w, b, a);
endmodule
```

→ `assign #(3,5) w = a & b;`

`w = a & b | c & d`

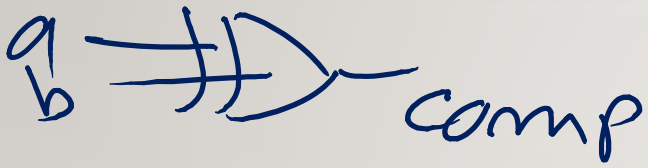
a	b	w
0	0	1
0	1	1
1	0	1
1	1	0

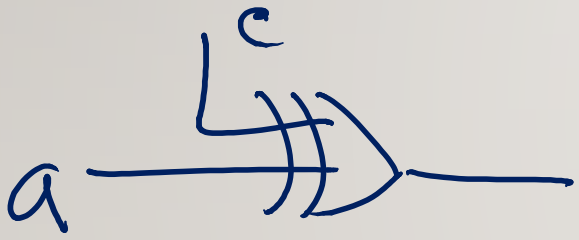
```
nand (w, a, b, c);

nand #(4) (w, a, b, c, d);

nand #(3,5) (w, a, b);

assign #(3,5) w = ~(a & b);
```

A & B

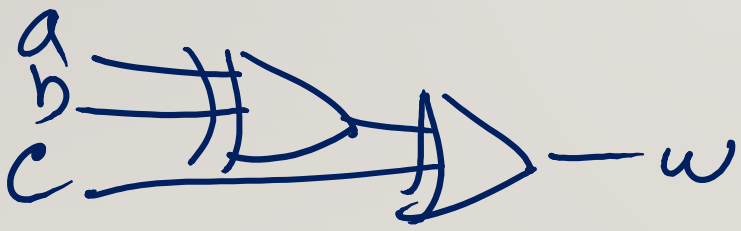
w = ~(a & b & c & d);

a	b	w
0	0	0
0	1	1
1	0	1
1	1	0





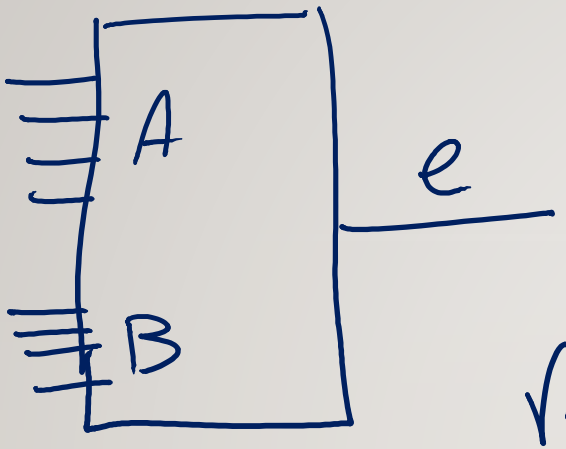
controlled inv.




```
xor (w, a, b, c);
```

```
assign w = a ^ b;
```

12

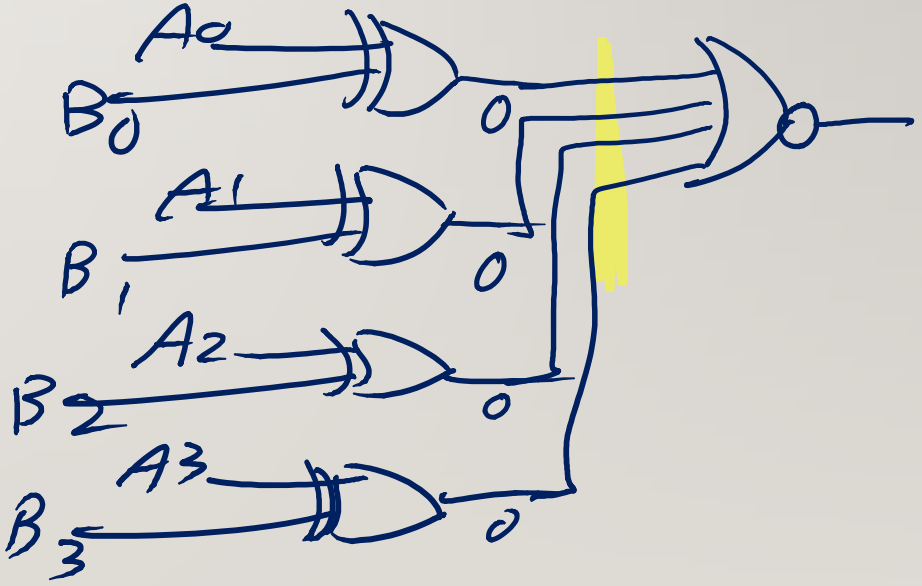


reduction

assign e = ~^(A ^ B)

& vector

Bitwise



```
module simpleComp(
    input [3:0] A, B,
    output e);

  assign e = ~|(A^B);
endmodule
```

Codes (lecture 1B)

Binary (Int, Frac, signed)

BCD

Gray

ASCII

15

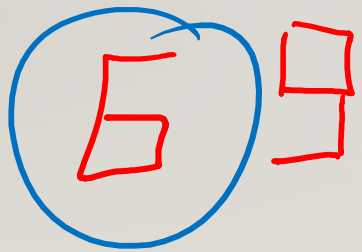
BCD: Binary Coded Decimal

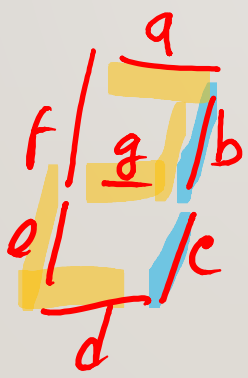
508-831-5555

0101

00001000 1000 0011 0001 0101 0101 0101 0101

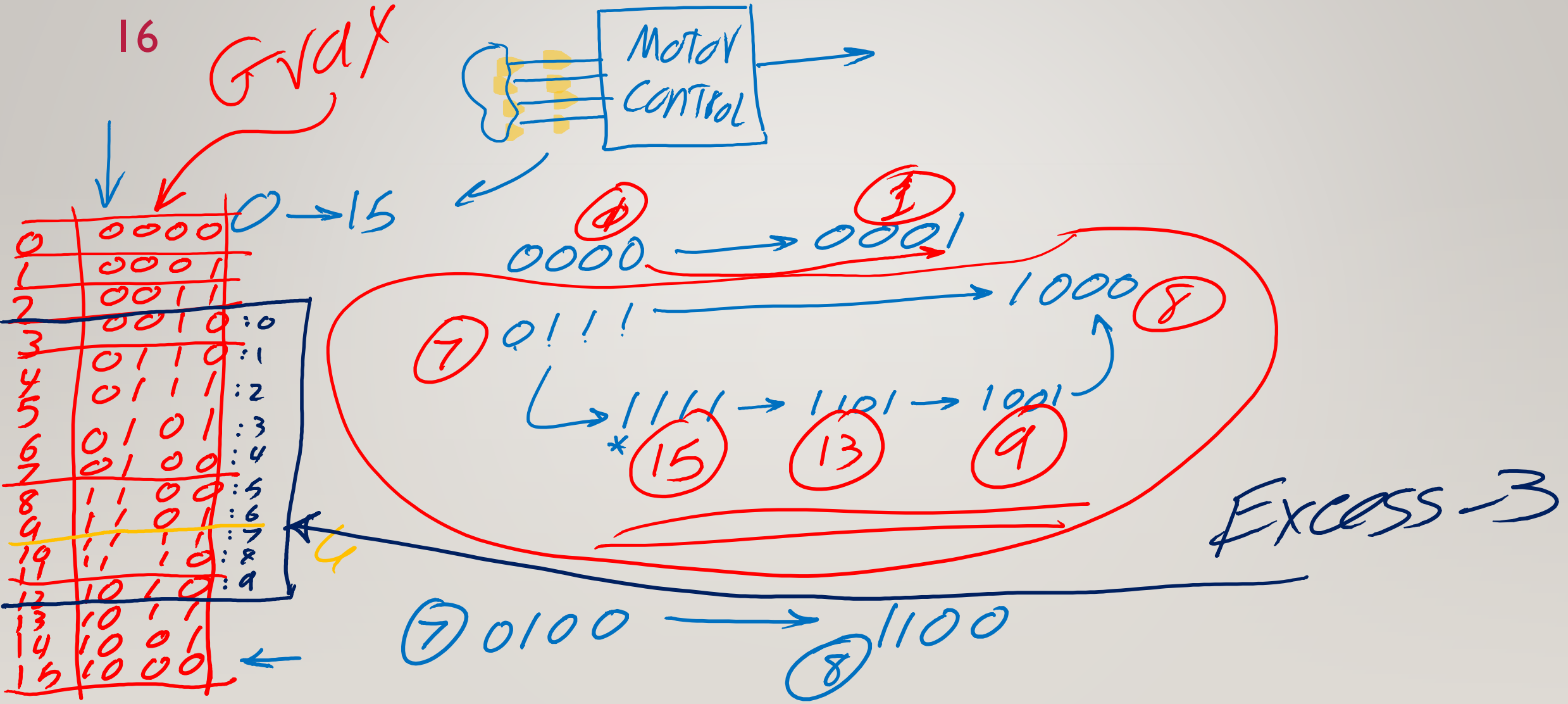
SSD





	a	b	c	d	e	f	g
0:	1	1	1	1	1	1	0
1:	0	1	1	0	0	0	0
2:	1	1	0	1	1	0	1
3:	1	1	1	1	0	0	1
4:	0	1	1	0	0	1	1
5:	1	0	1	1	0	1	1
6:	1	0	1	1	1	1	1
7:	1	1	1	0	0	0	0
8:	1	1	1	1	1	1	1
9:	1	1	1	1	0	1	1

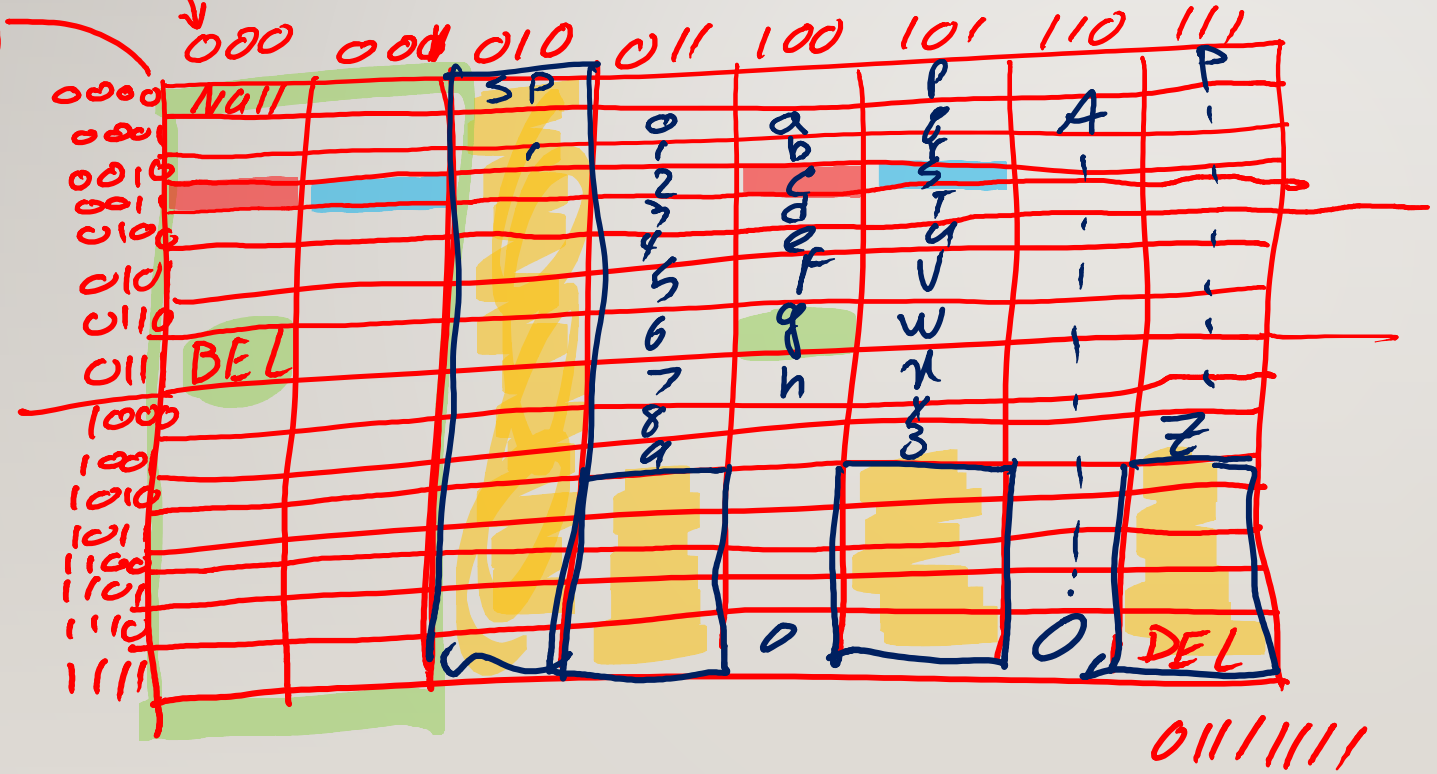
0000: 0
0001: 1
0010: 2
0011: 3
0100: 4
0101: 5
0110: 6
0111: 7
1000: 8
1001: 9


16
Gray
Motor
Control
0 → 15
0000
0001
1000
0111
1111
1101
1001
15
13
9
Excess-3
0100
1100

ASCII, extended

MSB (6) ●0000000 10000000 ü, ñ, ..

LSB (0)



01000000

Control @

Control C

Control 5