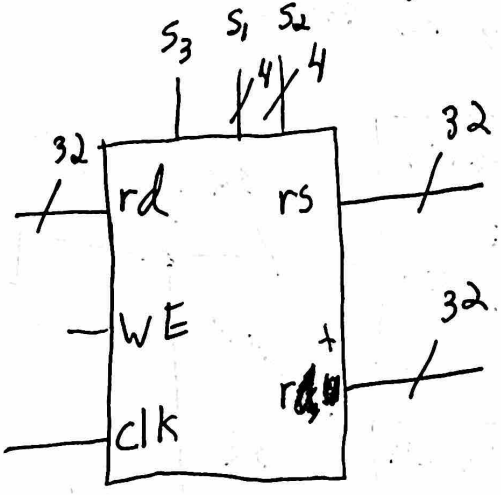
32 Bit register



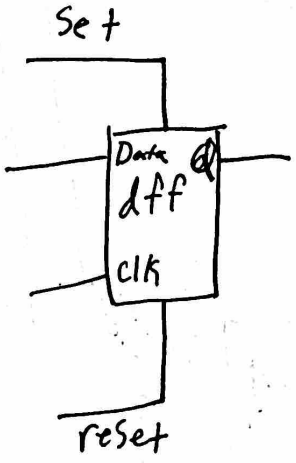
input or output data
- rt - source register
- rs - 2nd source register
- rd - destination register
- clk - clock
- S_1 - Select For Rs
- S_2 - select For Rt
- S_3 - Select For Rd
- WE global write enable

0 1
1 2
2 4
3 8
4 16
32

4 Bits need to specify any one of 32 registers

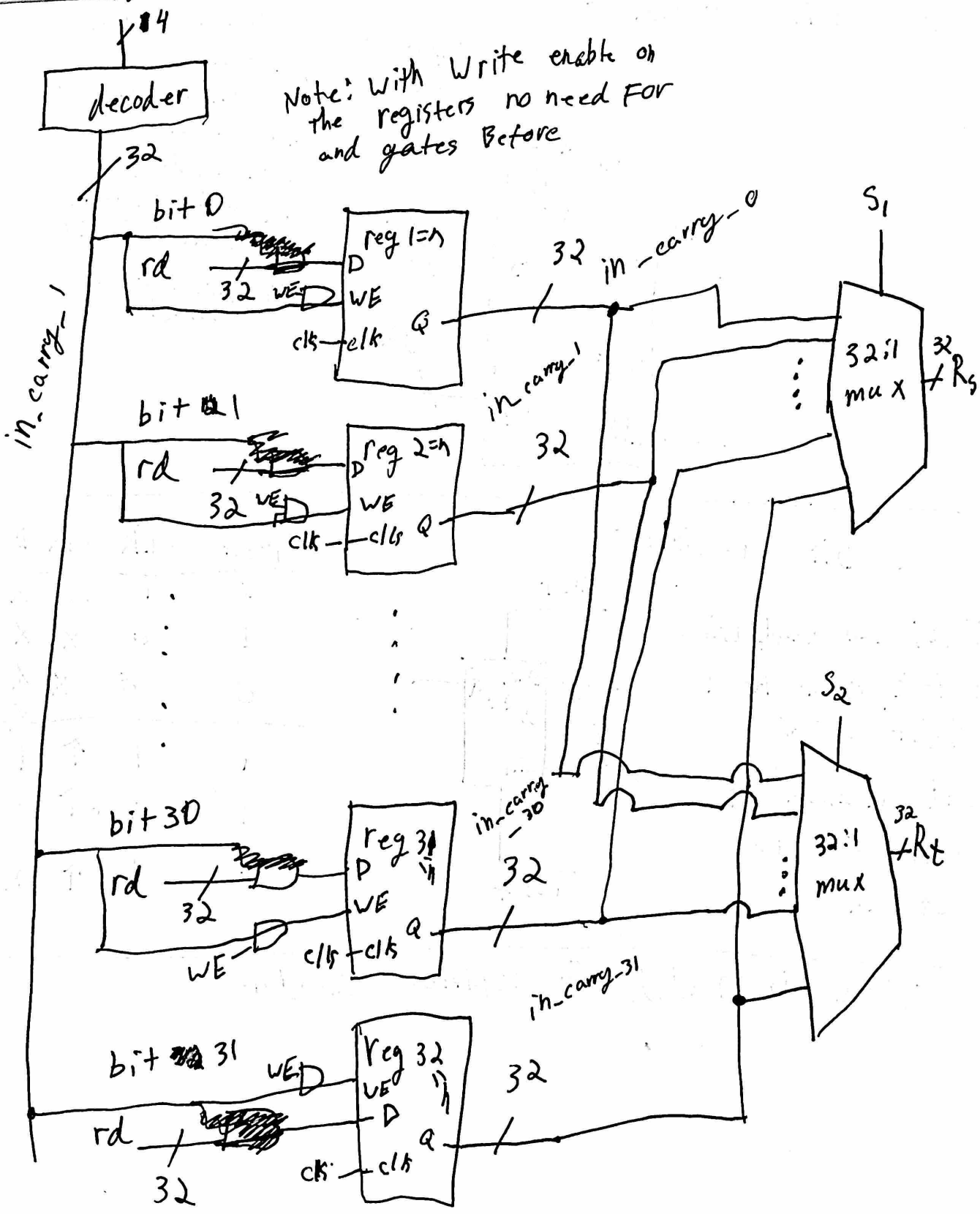
1 Bit register

Note: can hard wire set & reset to 1 right now and can use them later or not at all



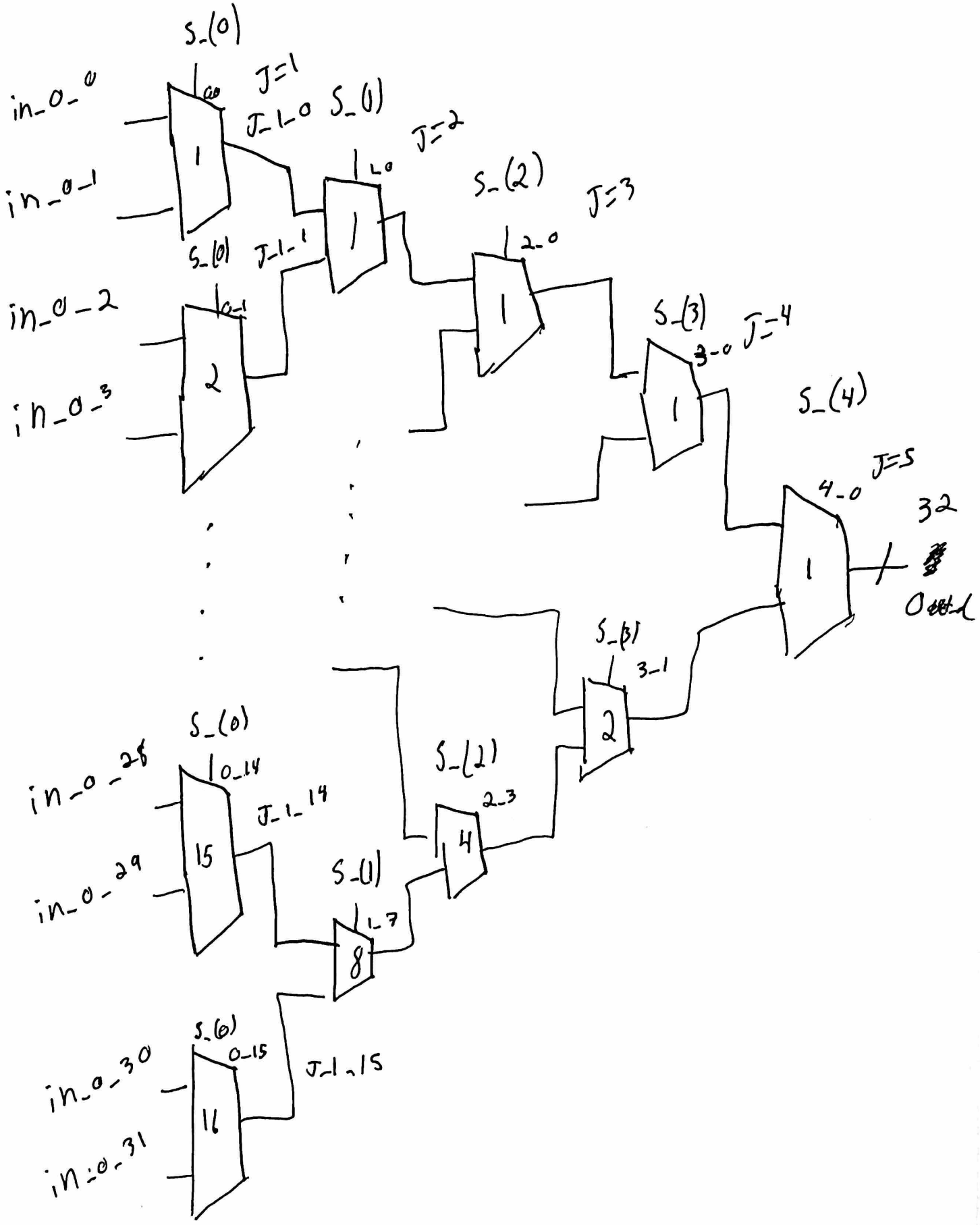
reset	CLR	clk	D	Q
0	1	X	X	1
1	0	X	X	0
0	0	X	X	X X
			X	
1	1	↑	1	1
1	1	↑	0	0
1	1	↑	X	Q_0

4:32 is needed For register Decoder

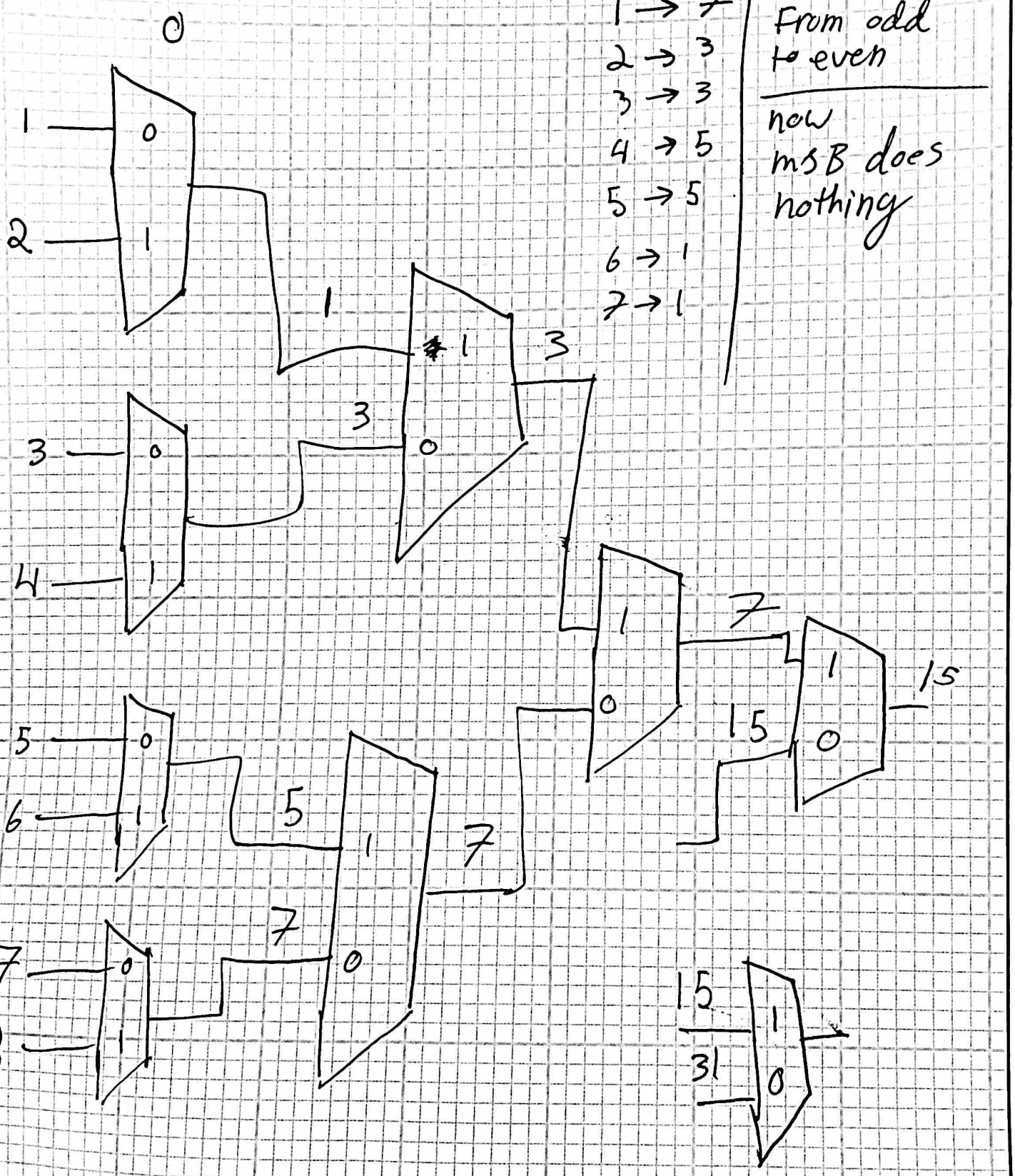

5
decoder
Note: With Write enable on the registers no need For and gates Before
32
bit 0
rd
32
WE
D
WE
clk
clk
reg 1=n
Q
32
in_carry_0
S1
in_carry_1
32:1
mux
32
Rs
bit 1
rd
32
WE
D
WE
clk
clk
reg 2=n
Q
in_carry_1
32
S2
bit 30
rd
32
WE
D
WE
clk
clk
reg 31
Q
in_carry_30
32
32:1
mux
32
Rt
in_carry_31
bit 31
WE
rd
32
WE
D
clk
clk
reg 32
Q
32
in_carry_1

32:1 mux

16+8+4+2+1 internal carries



can I reverse the inputs of the first array



0 → 7
1 → 7
2 → 3
3 → 3
4 → 5
5 → 5
6 → 1
7 → 1

msB changes from odd to even

now msB does nothing

FLINT HILLS resources	TITLE:			PROJECT NO.
	PROJECT:	BY	DATE	REV.

internal carry of register n

internal_carry_n corresponds to

16 isn't working

look at running a vhdl process this lets you write out a consecutive

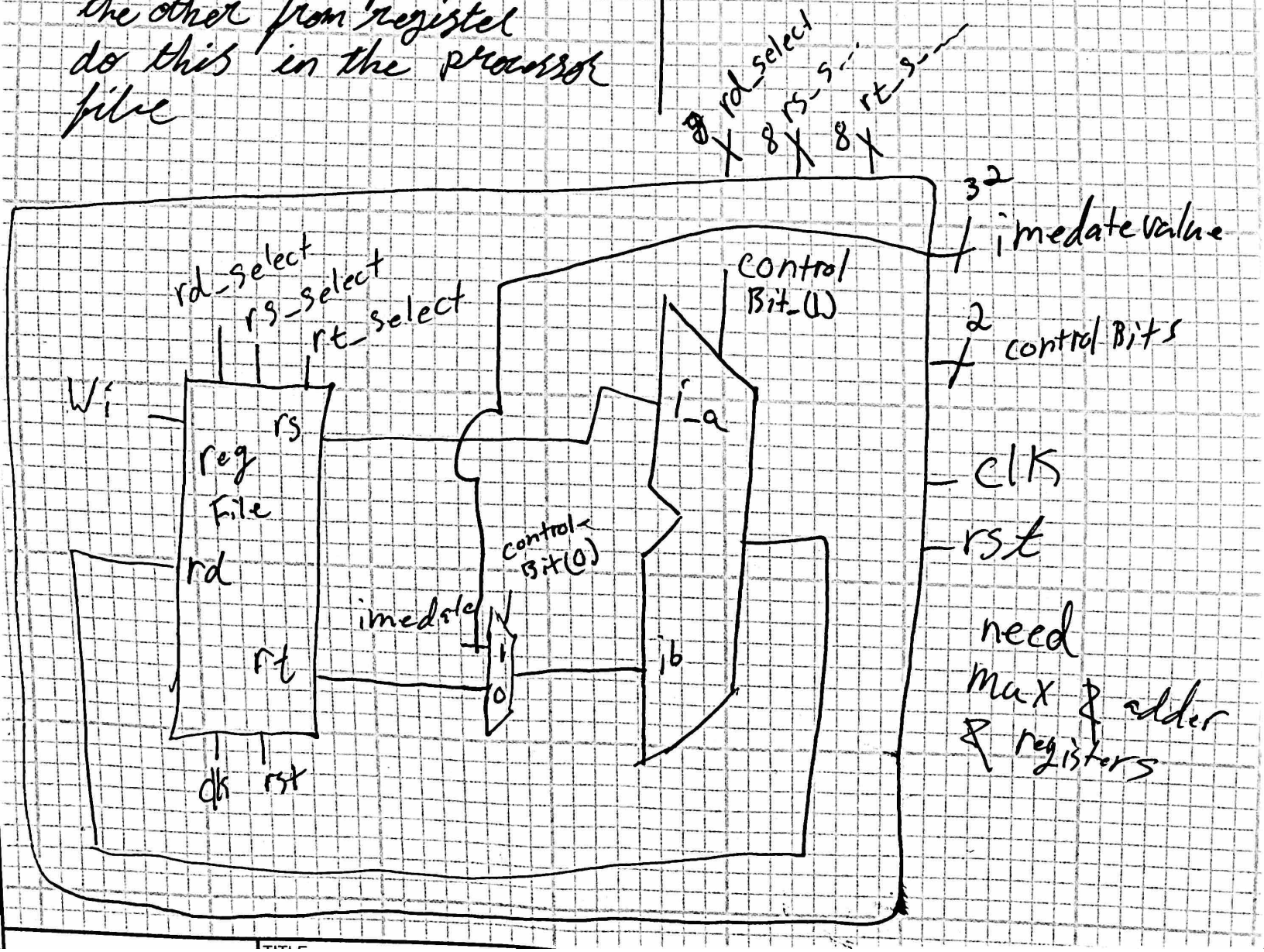
- add the mux to the add subtracter
  - one as an input the other from register
  do this in the processor file

rd_select = place to store value

rs_select = register that is always put into ALU

rt = register when adding 2 together

0 - add
1 - addi
2 - sub
3 subi



32
rd
1
wi
Reg
File
rt
rs
5
5
5
rd_select
rs_select
rt_select
32
immediate
val
c+1_B
2
c+1(0)
c+1(1)
G
-
ib
ia