CSEE 3827: Fundamentals of Computer Systems, Spring 2022

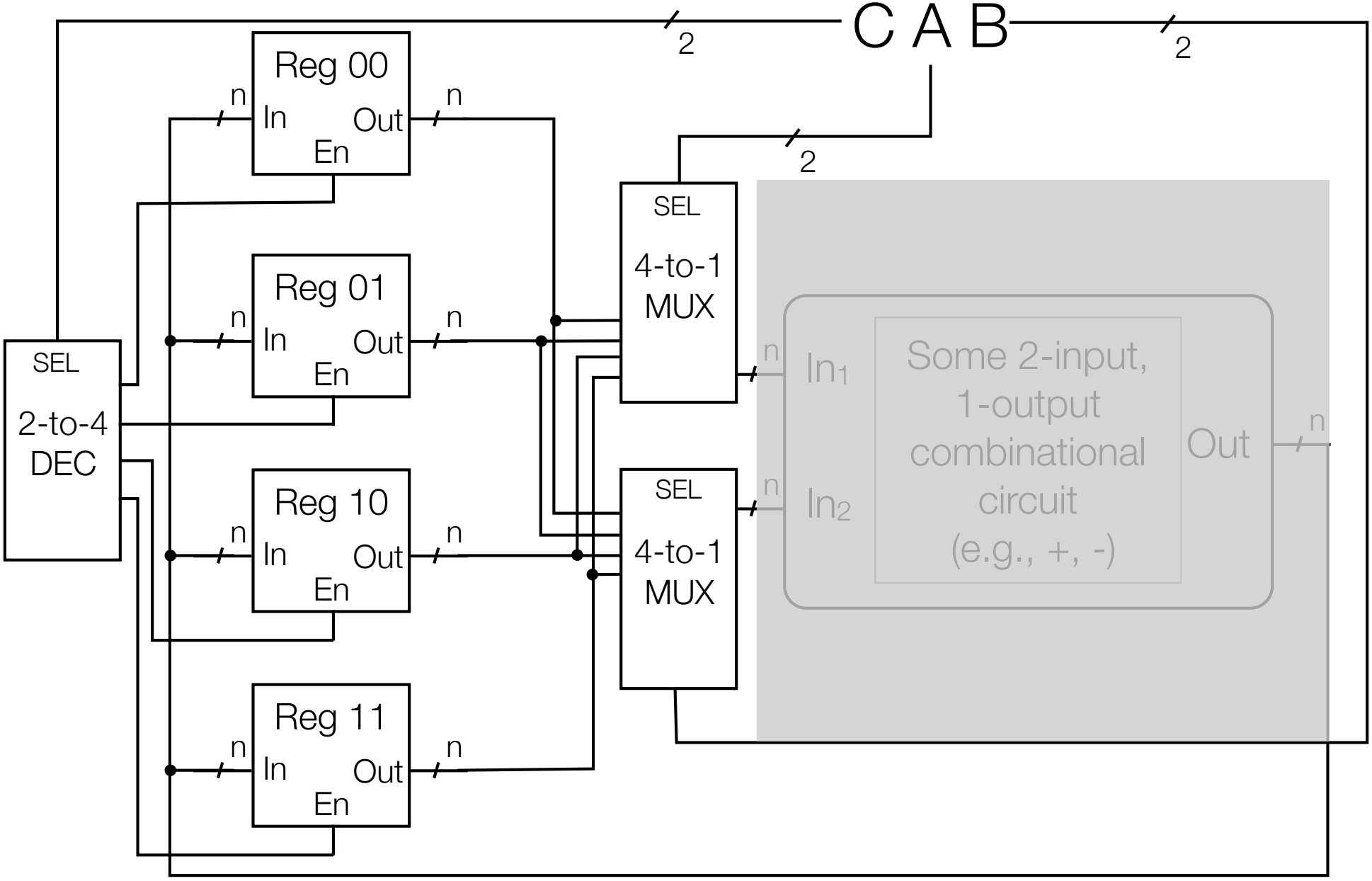
Lecture 9

Prof. Dan Rubenstein (danr@cs.columbia.edu)

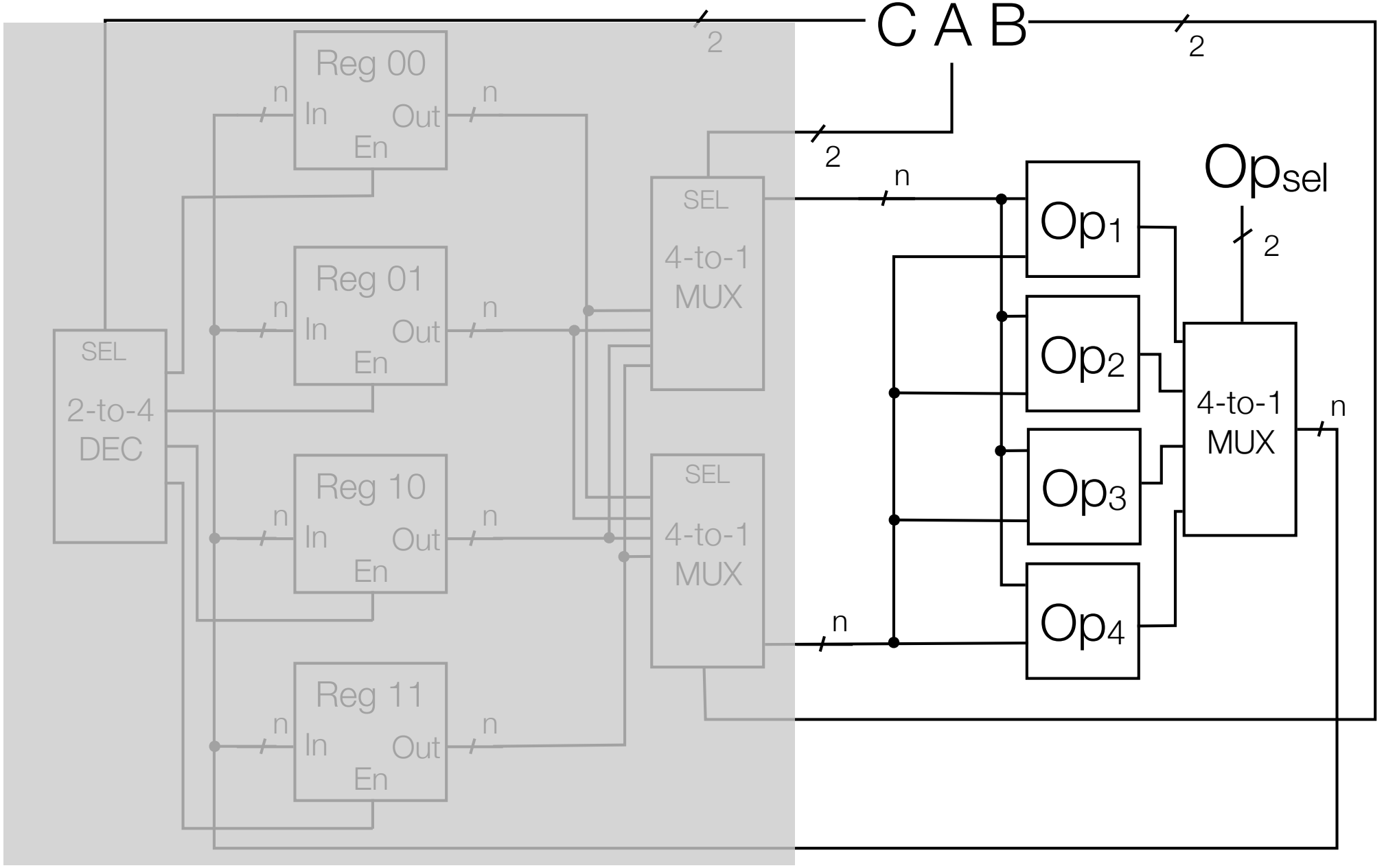
Agenda (M&K 9.1-9.8)

- Assembly Language: Very high level overview
- Computer Design Basics
  - Datapath
  - Function Unit
    - Arithmetic/Logic Unit (ALU): Multiplexers galore!
    - Shifter (revisited quickly)
  - Control Word
  - Instruction & Instruction Decoder

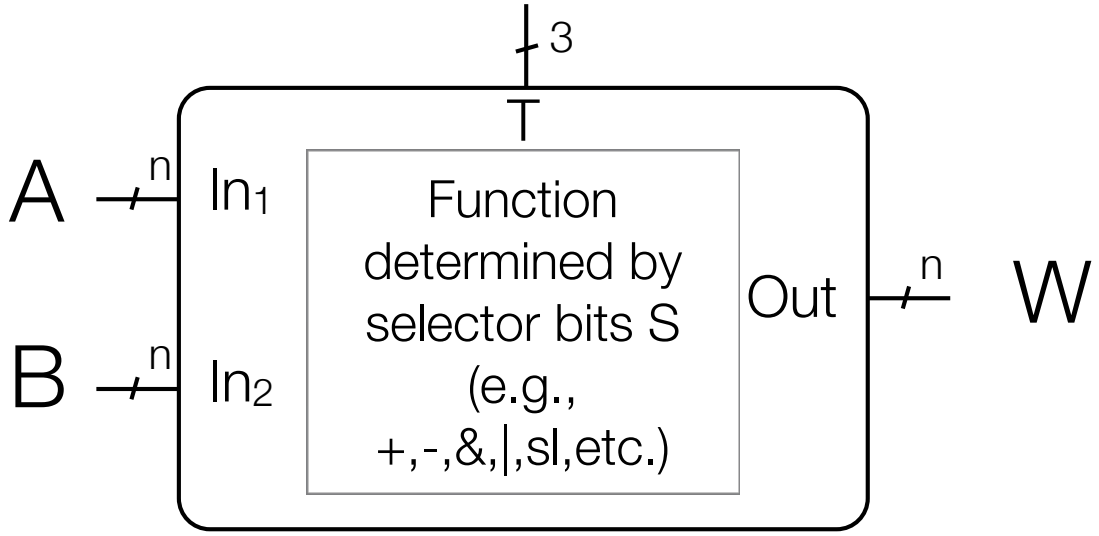
Small Register file Example



Function Unit (also seen before, Lecture 6)



(Simplified) MIPS Function Unit at a high level



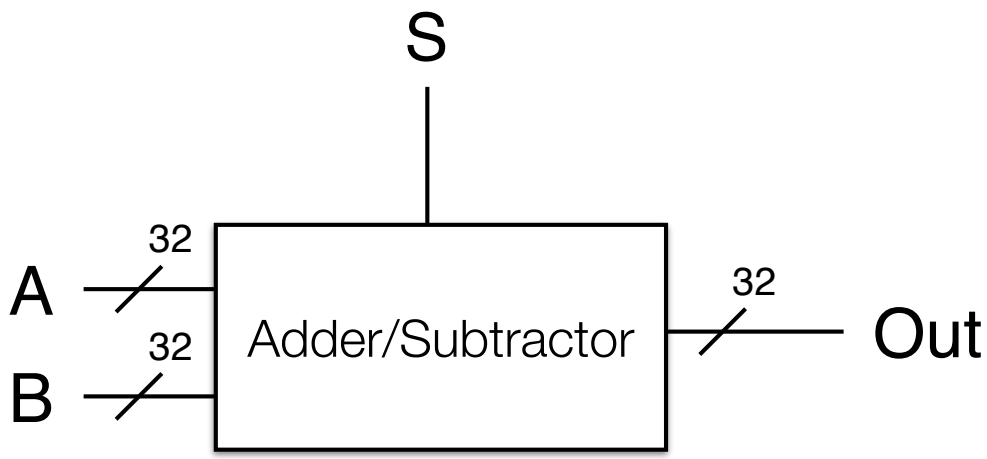
- Combinational Circuitry
- OP is the operation determined by 3-bit selector T

Arithmetic Logic (function) Unit

- Goal: Build a **combinational circuit** that has the following functionality (on 32-bit inputs A & B) with a 3-bit selector T=XYZ:

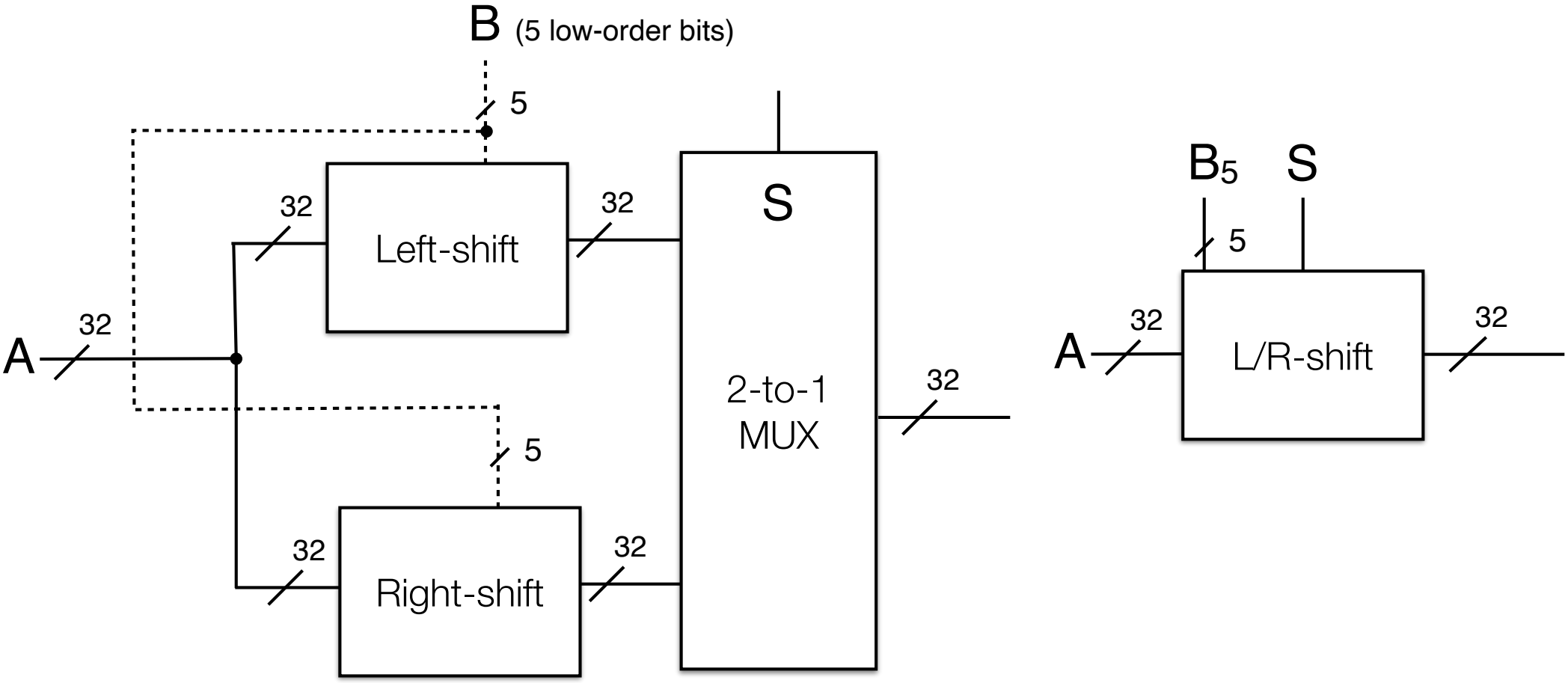
T=XYZ	Output	Comments
“000”	A + B	Add
“001”	A - B	Subtract
“010”	sl A ← B	Shift left by 5-bit low order bits of B
“011”	sr A ← B	Shift right by 5-bit low order bits of B
“100”	A B	logical AND
“101”	A + B	logical OR
“110”	A ⊕ B	logical XOR
“111”	$\overline{A}$	complement A (ignore B)

Adder/Subtractor



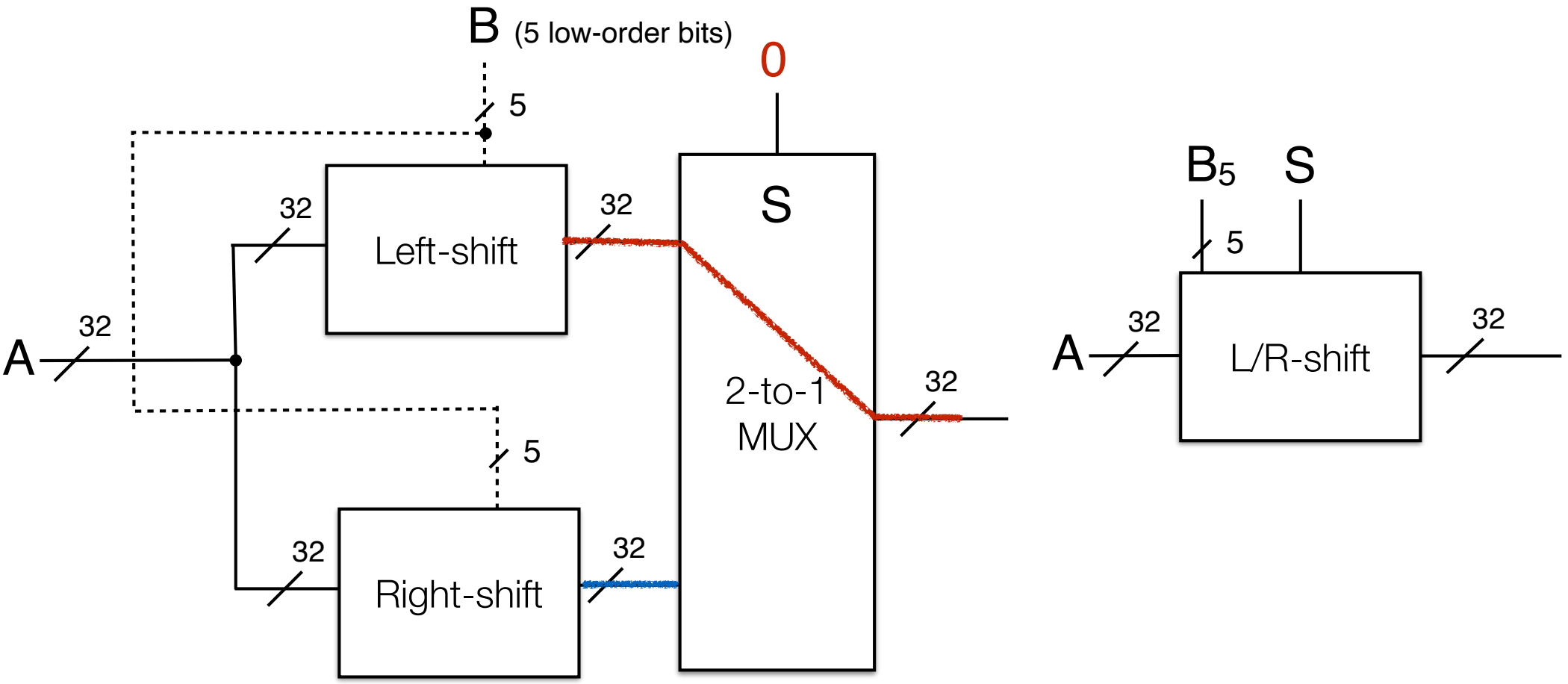
- Built earlier in class
- has a 1-bit Selector S. S=0 then Add A+B, S=1 then Subtract A-B
- Note: When XY=00, then want to use adder/subtractor, with Z feeding into S (so 000 does add, 001 does subtract)

L/R Shifter



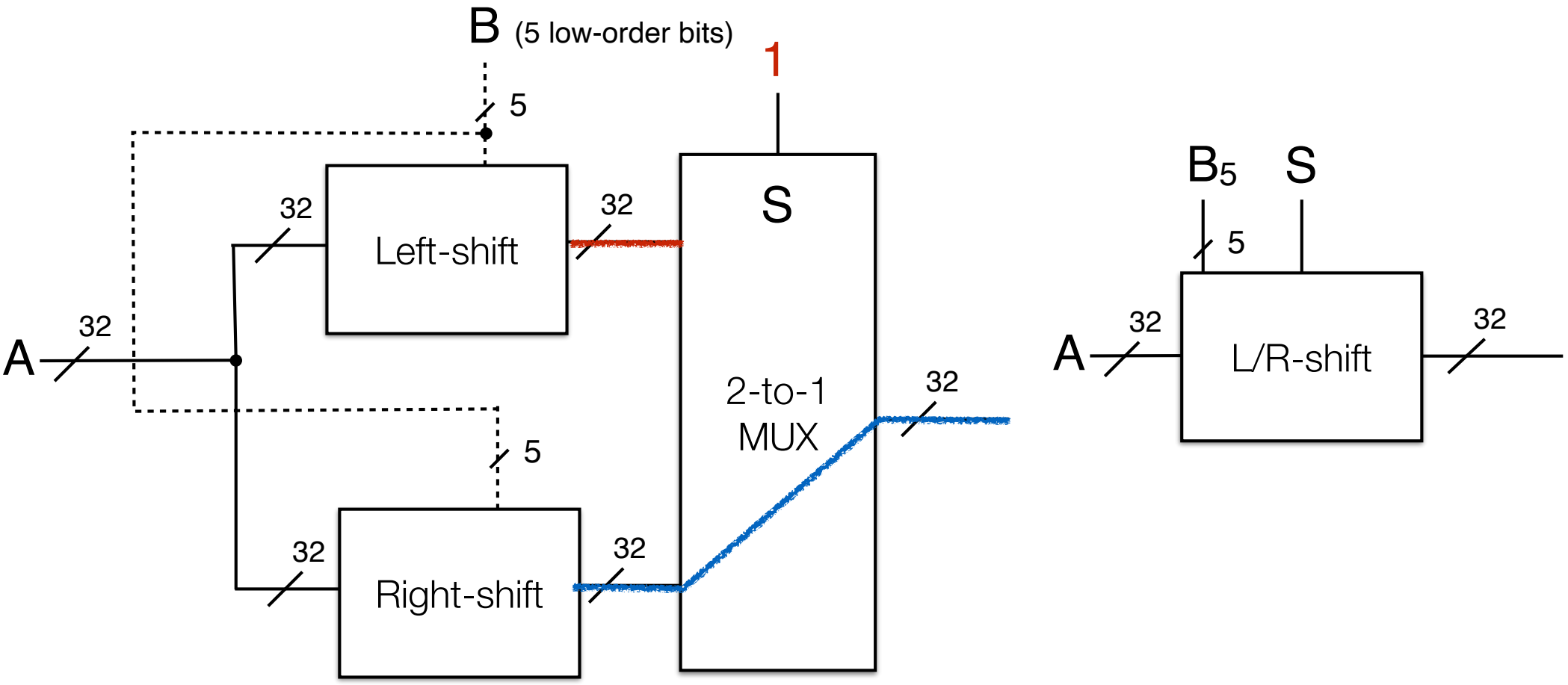
- Combine a Left-shift with a Right shift using a 2-to-1 MUX
- S=0 then shift A left by B bits, S=1 then shift A right by B bits
- When XY=01, want to use L/R shifter with Z feeding into S

L/R Shifter



- Combine a Left-shift with a Right shift using a 2-to-1 MUX
- S=0 then shift A left by B bits, S=1 then shift A right by B bits
- When XY=01, want to use L/R shifter with Z feeding into S

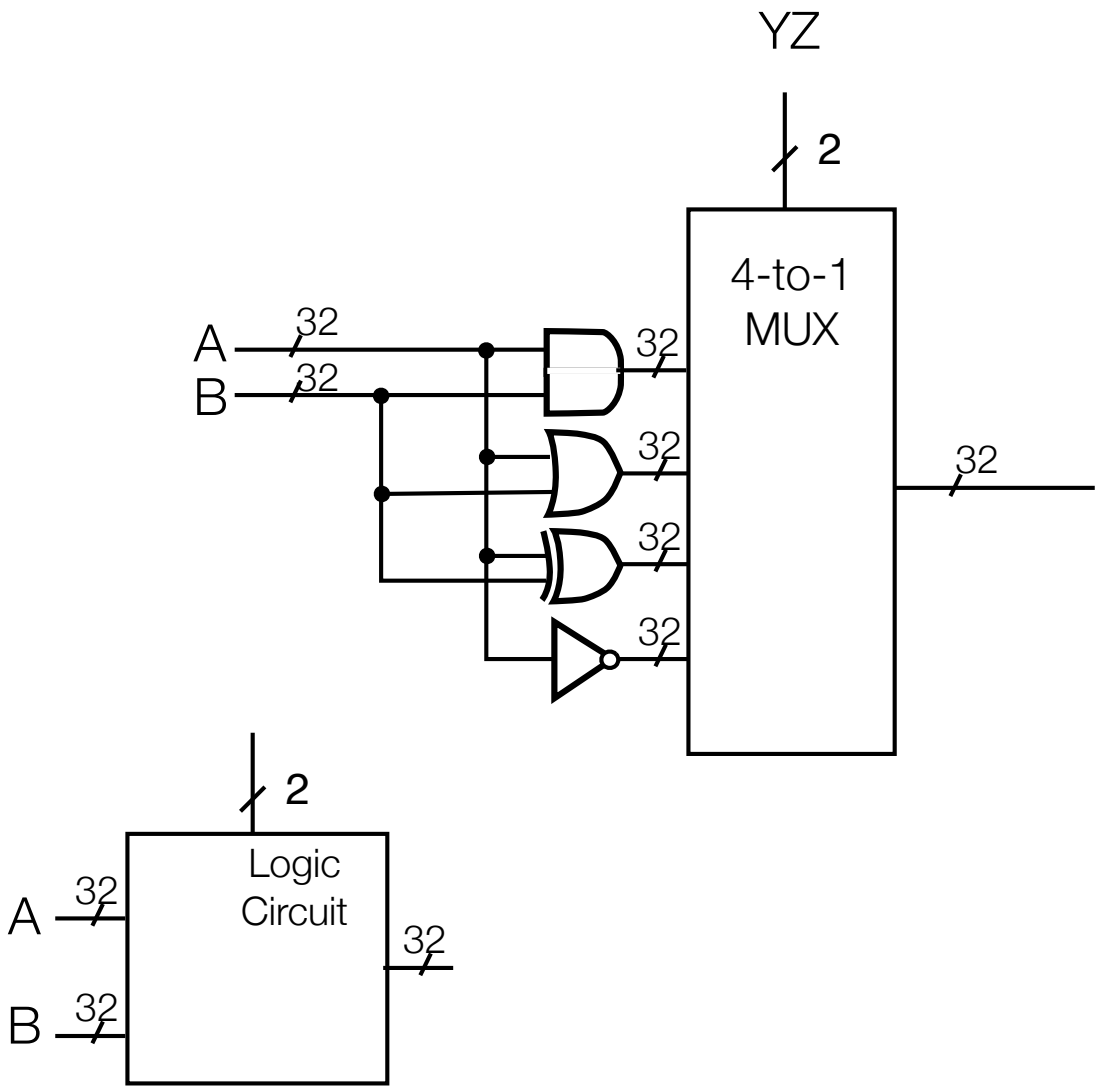
L/R Shifter



- Combine a Left-shift with a Right shift using a 2-to-1 MUX
- S=0 then shift A left by B bits, S=1 then shift A right by B bits
- When XY=01, want to use L/R shifter with Z feeding into S

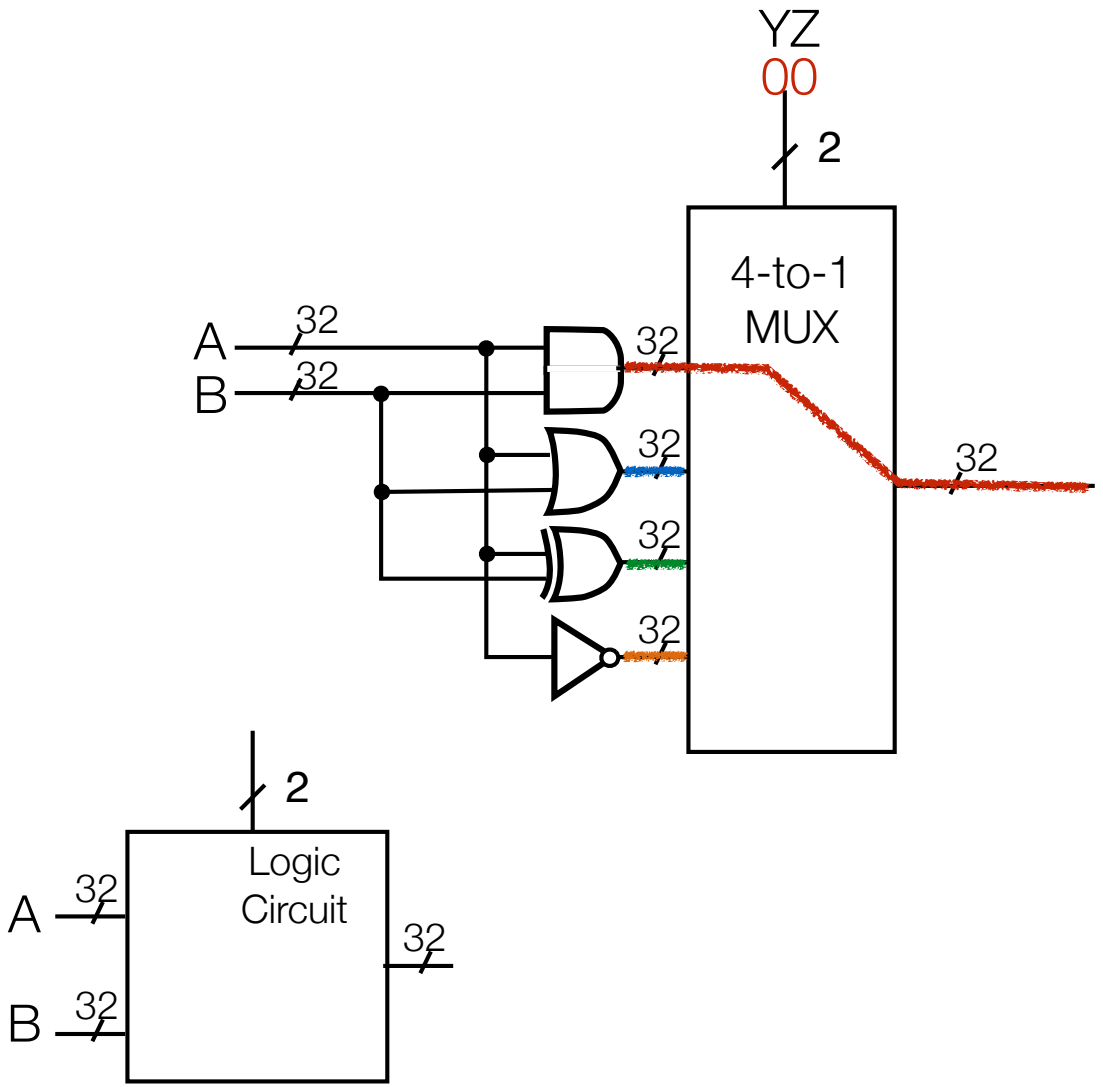
Step 2: Logic Circuit: As easy as it gets...

Y	Z	Op
0	0	AB
0	1	A+B
1	0	A⊕B
1	1	$\overline{A}$



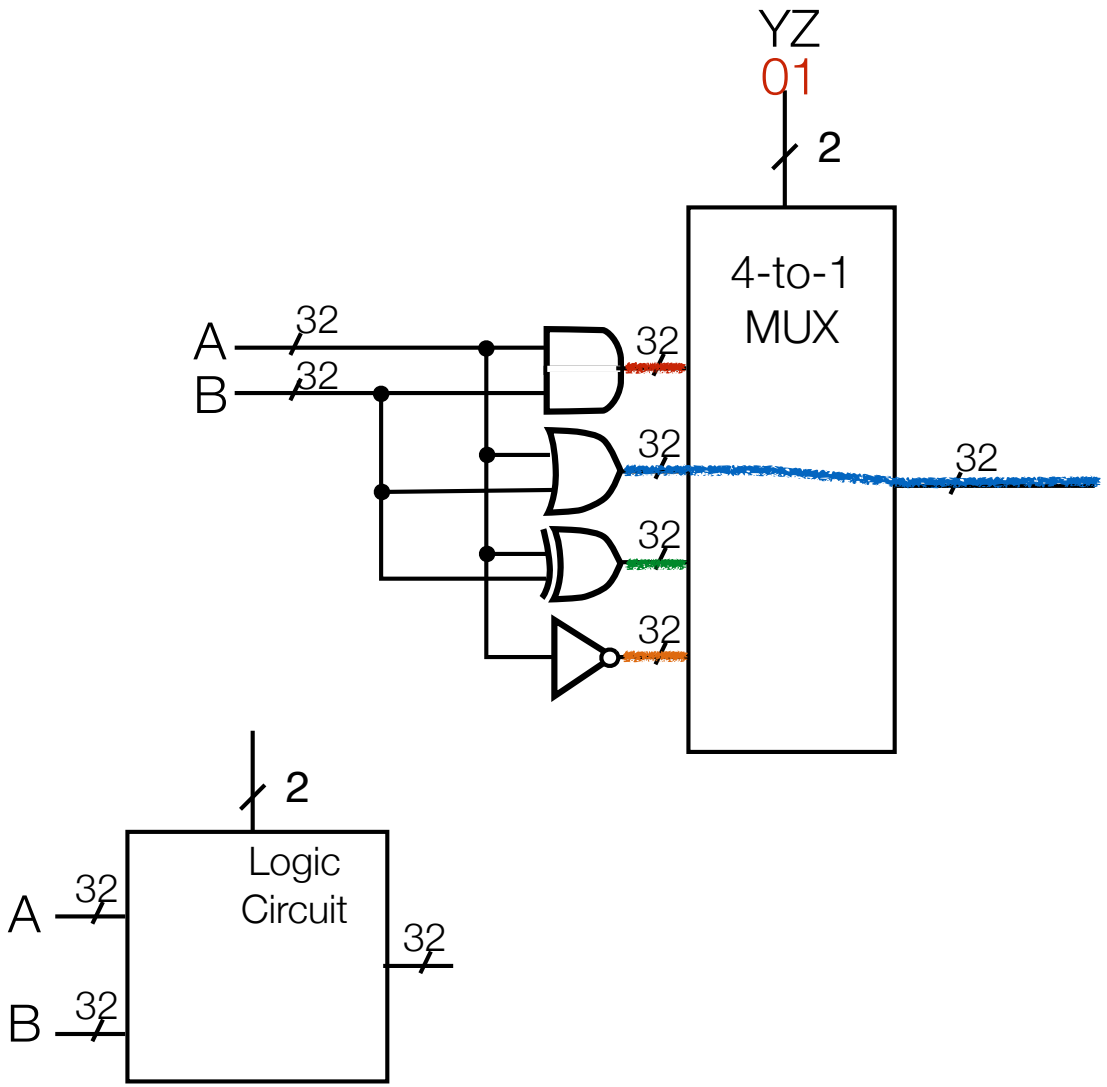
Step 2: Logic Circuit: As easy as it gets...

Y	Z	Op
0	0	AB
0	1	A+B
1	0	A⊕B
1	1	$\overline{A}$



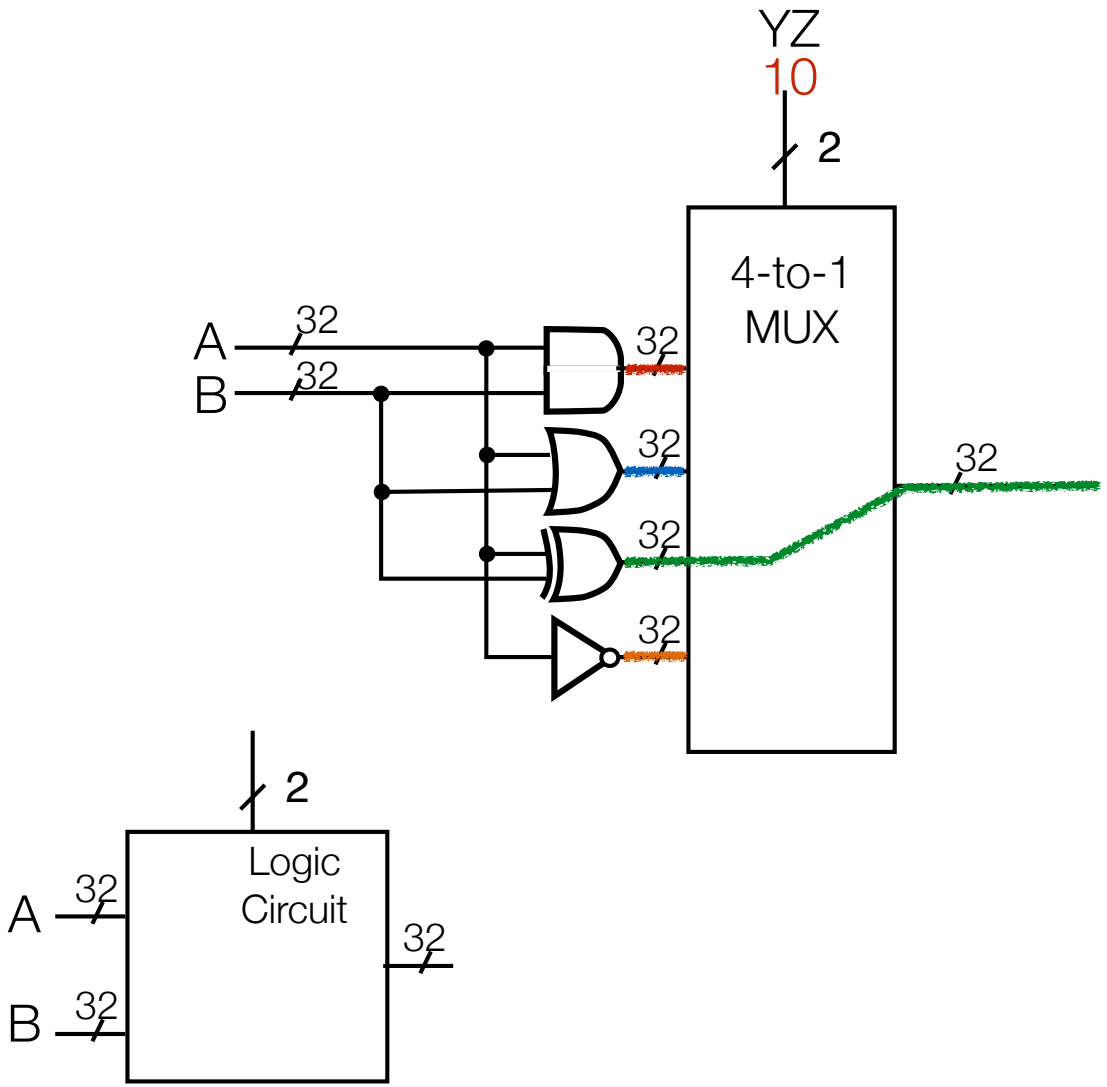
Step 2: Logic Circuit: As easy as it gets...

Y	Z	Op
0	0	AB
0	1	A+B
1	0	A⊕B
1	1	$\overline{A}$



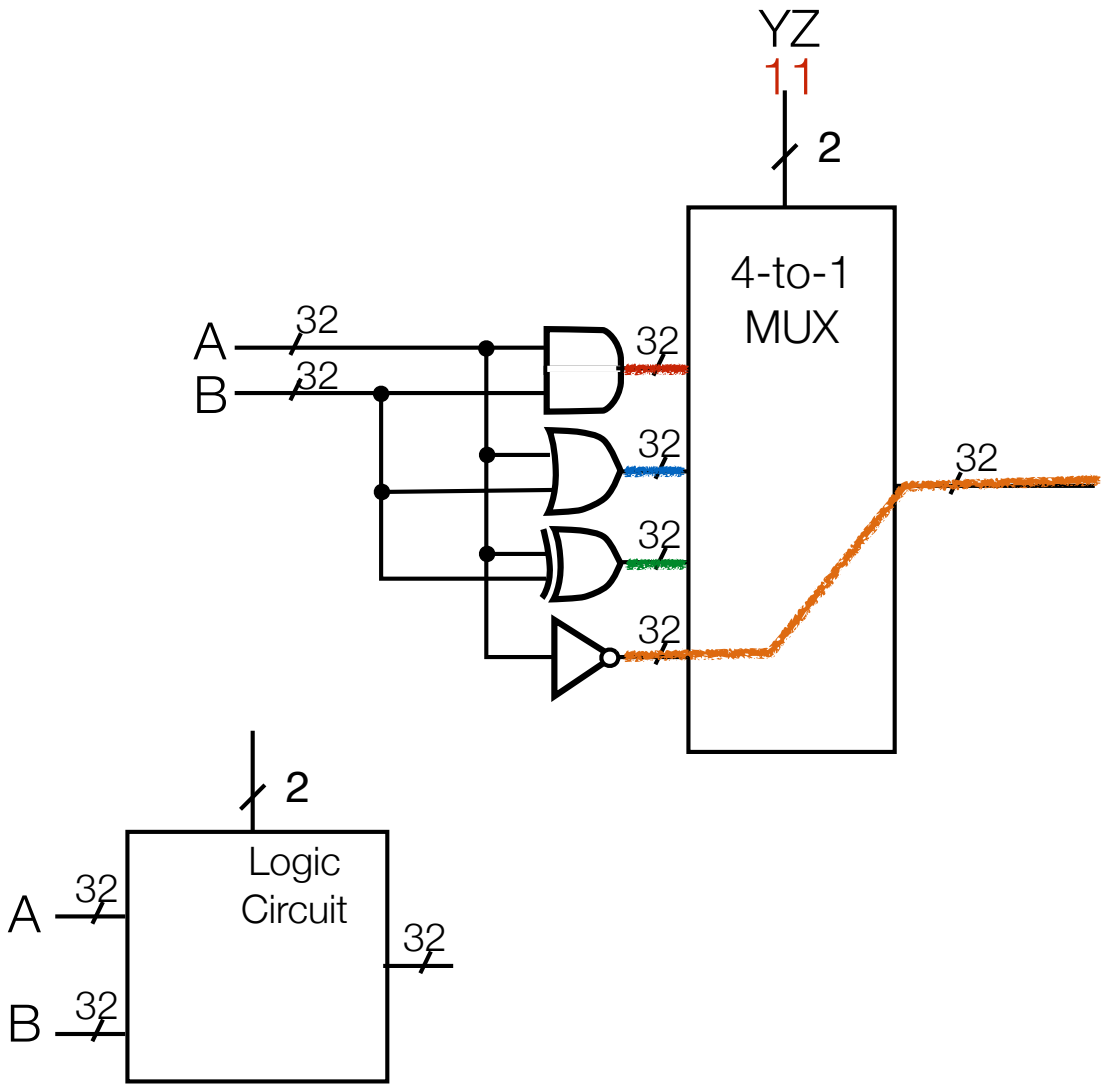
Step 2: Logic Circuit: As easy as it gets...

Y	Z	Op
0	0	AB
0	1	A+B
1	0	A⊕B
1	1	$\overline{A}$



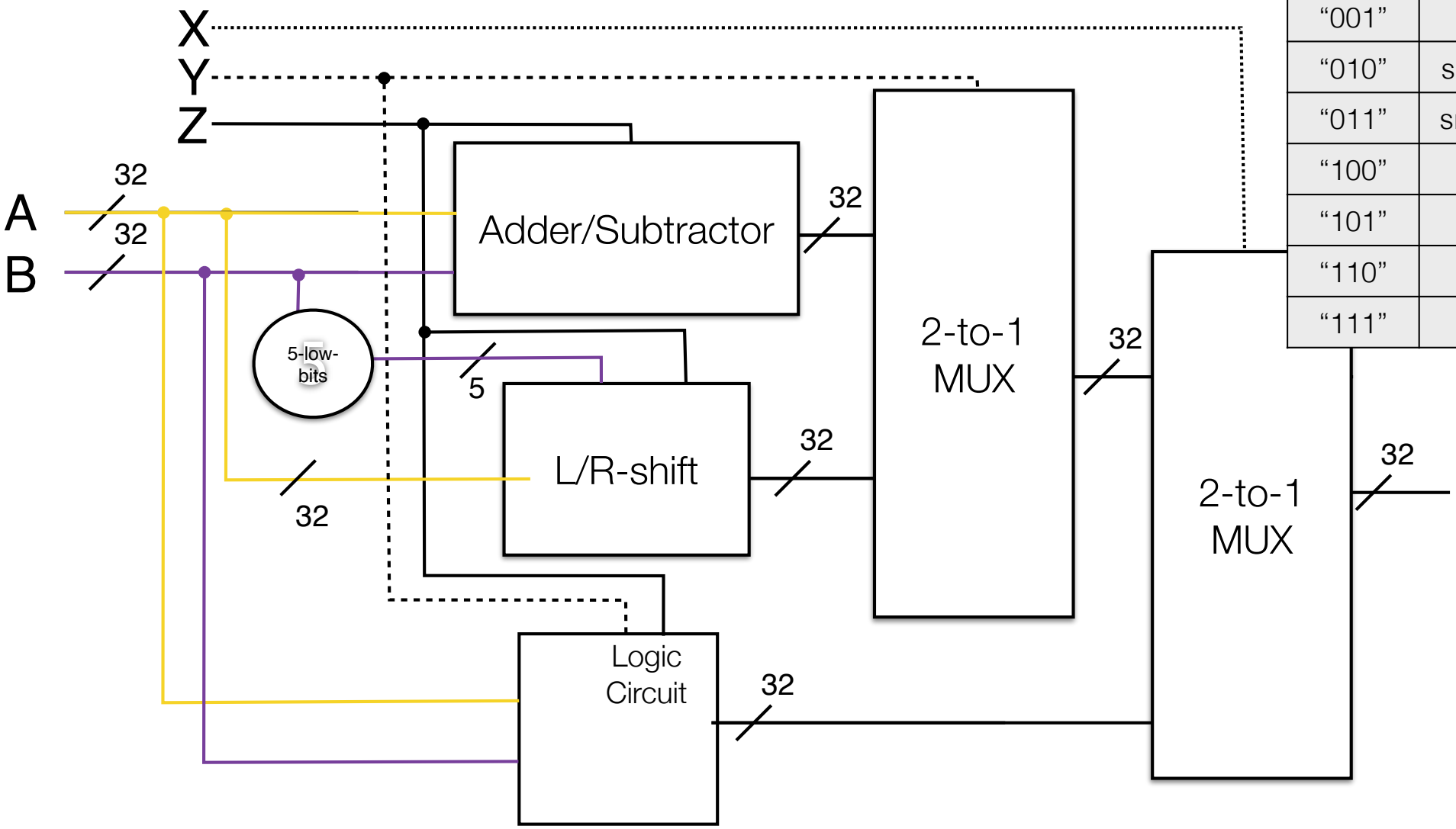
Step 2: Logic Circuit: As easy as it gets...

Y	Z	Op
0	0	AB
0	1	A+B
1	0	A⊕B
1	1	$\overline{A}$

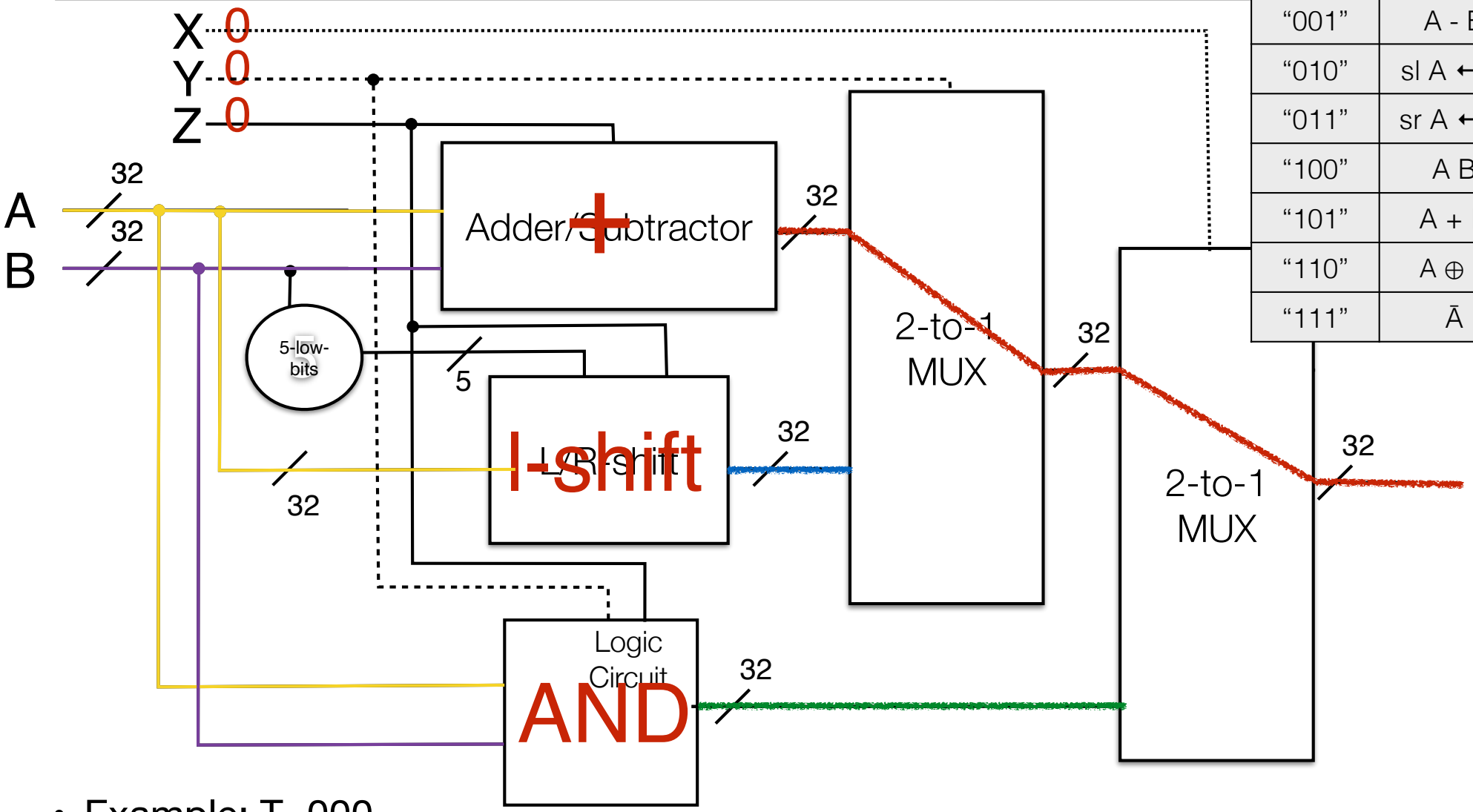


Putting it all together

T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā



Putting it all together

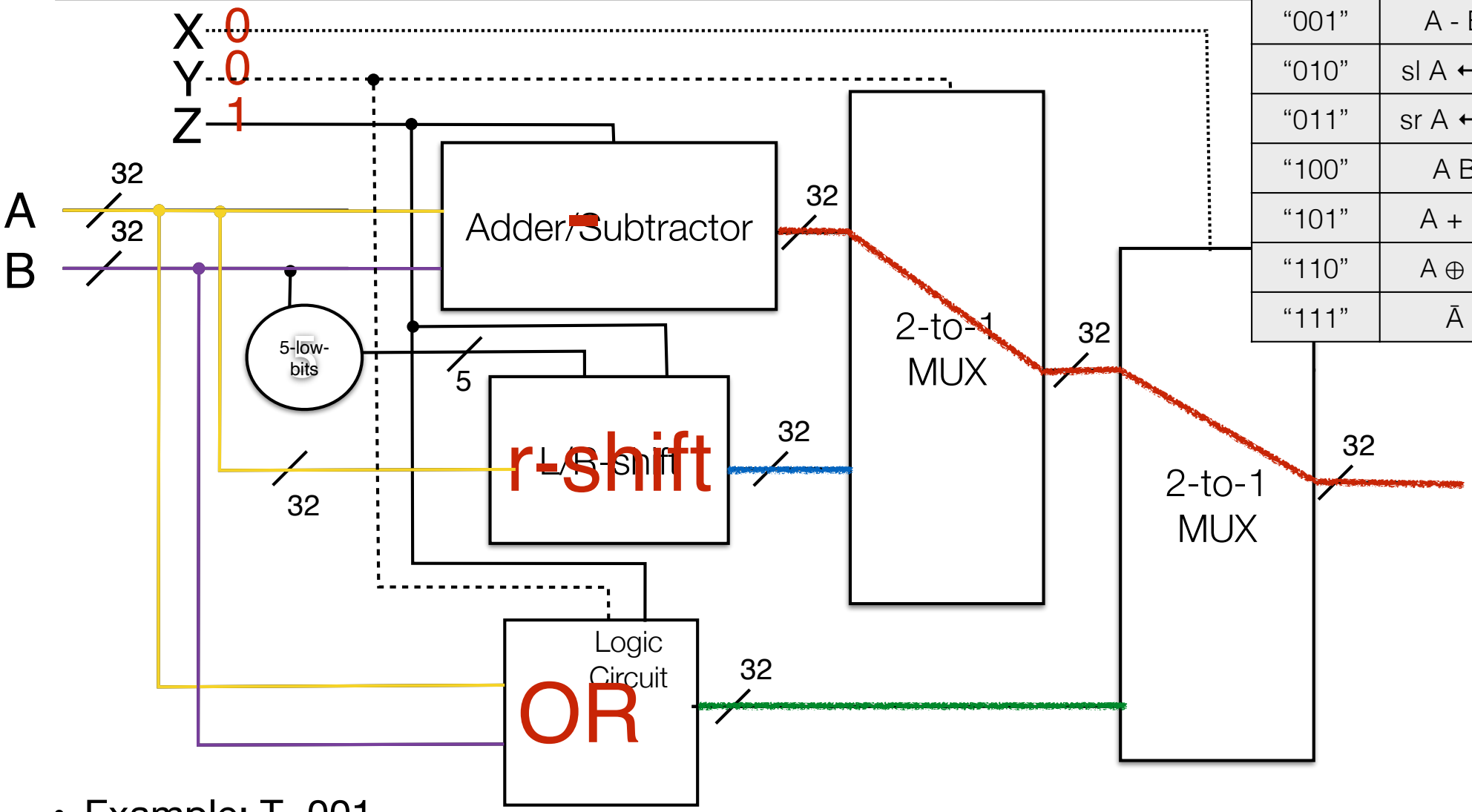


T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā

- Example: T=000

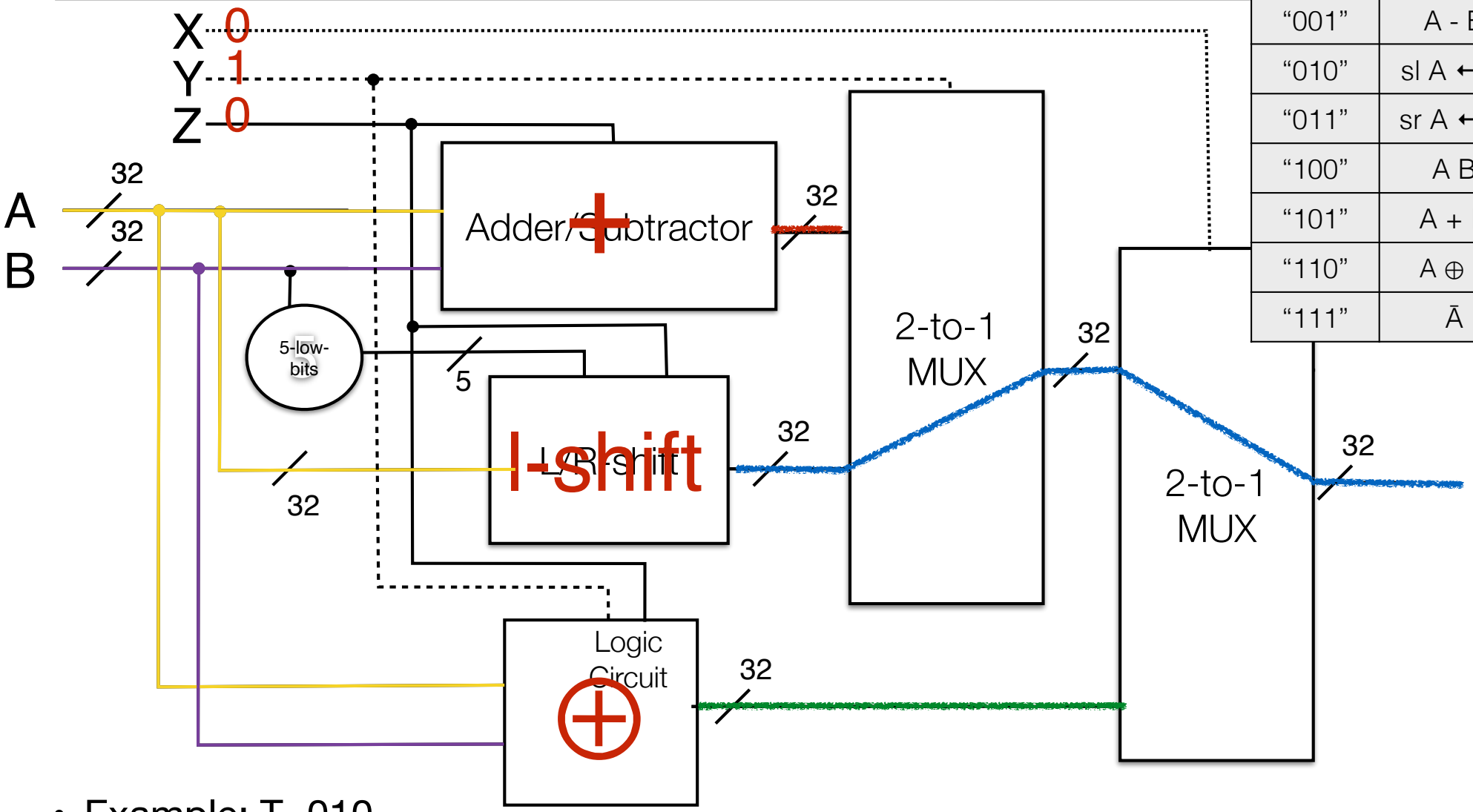
Putting it all together

T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā



- Example: T=001

Putting it all together

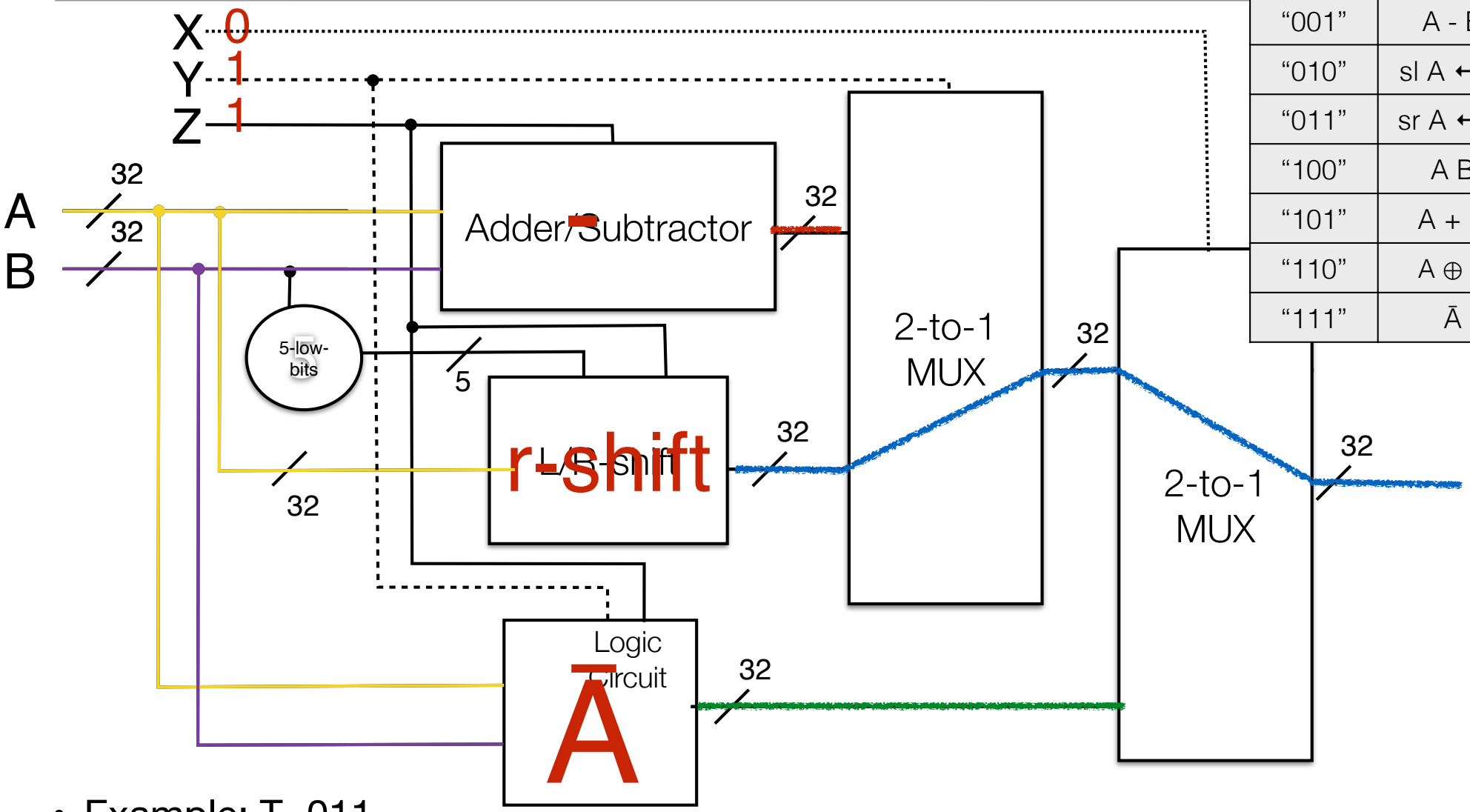


T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā

- Example: T=010

Putting it all together

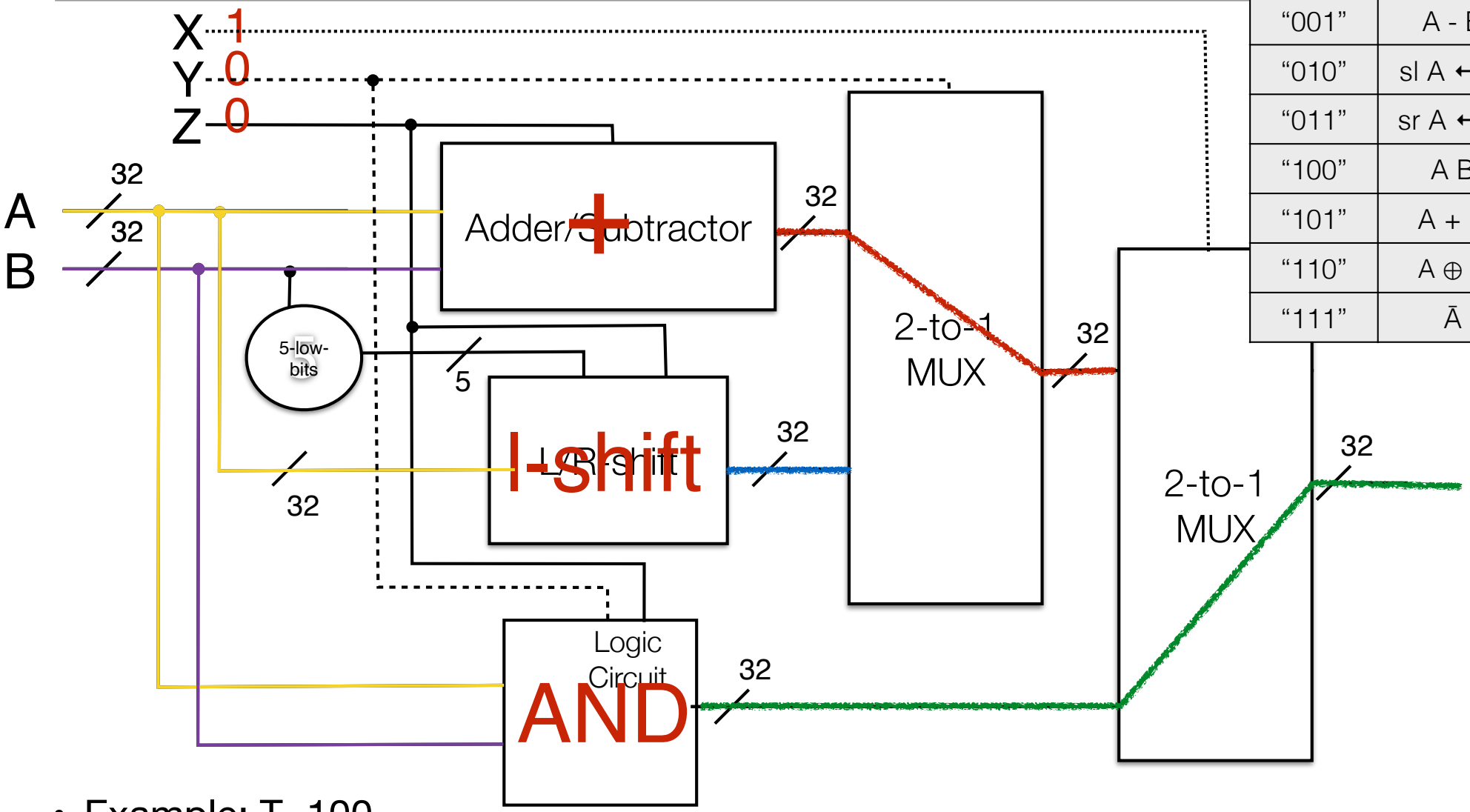
T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā



- Example: T=011

Putting it all together

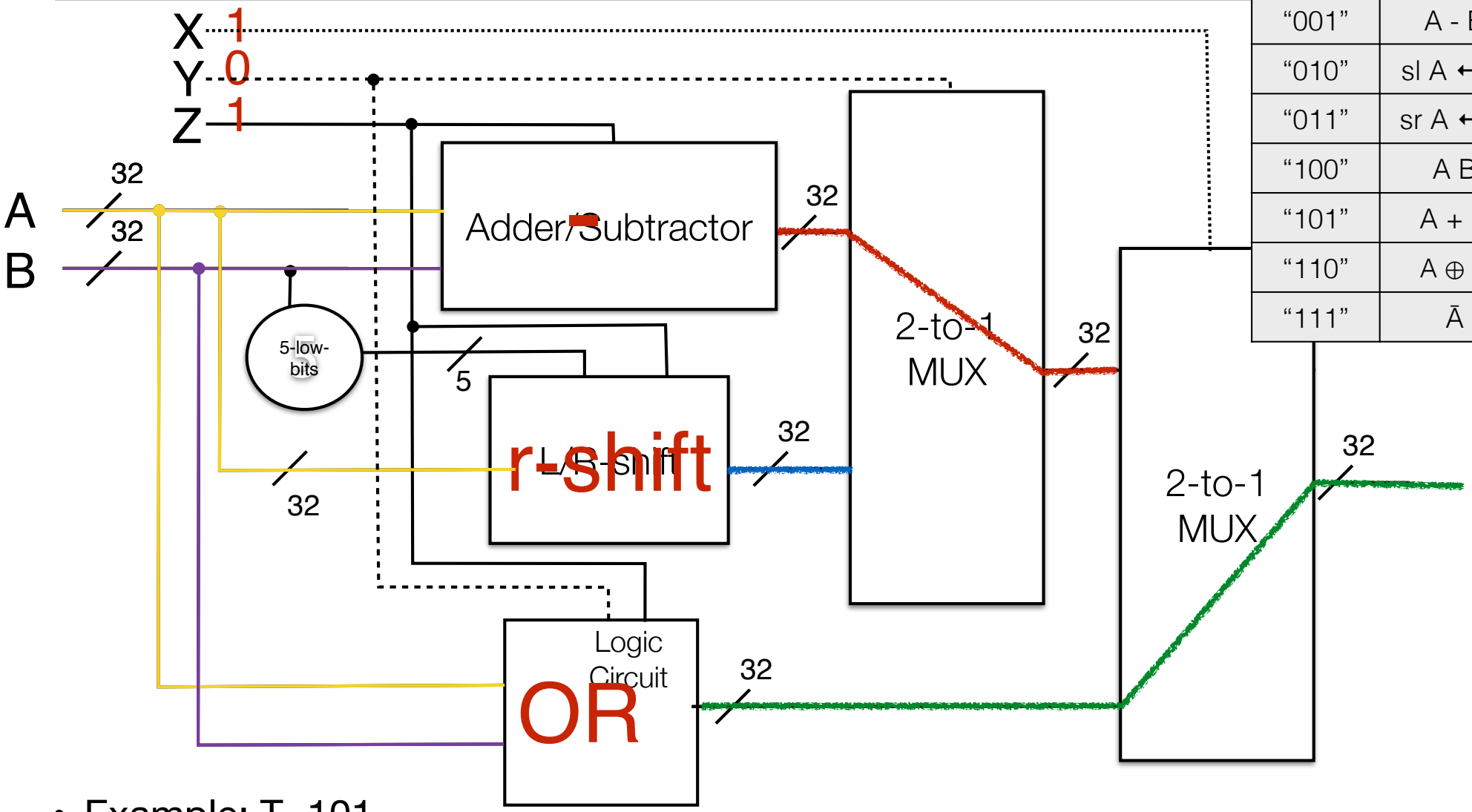
T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā



- Example: T=100

Putting it all together

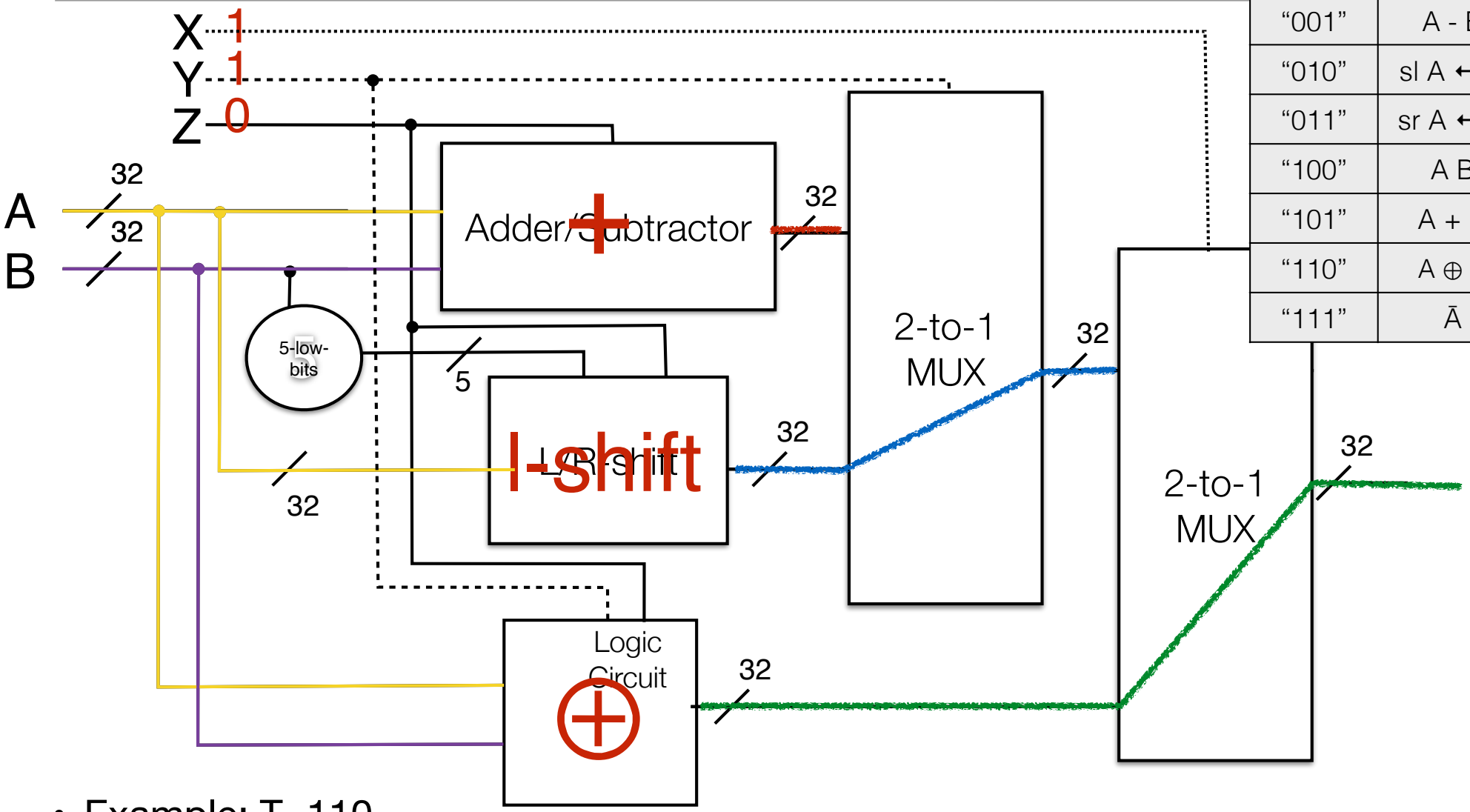
T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā



- Example: T=101

Putting it all together

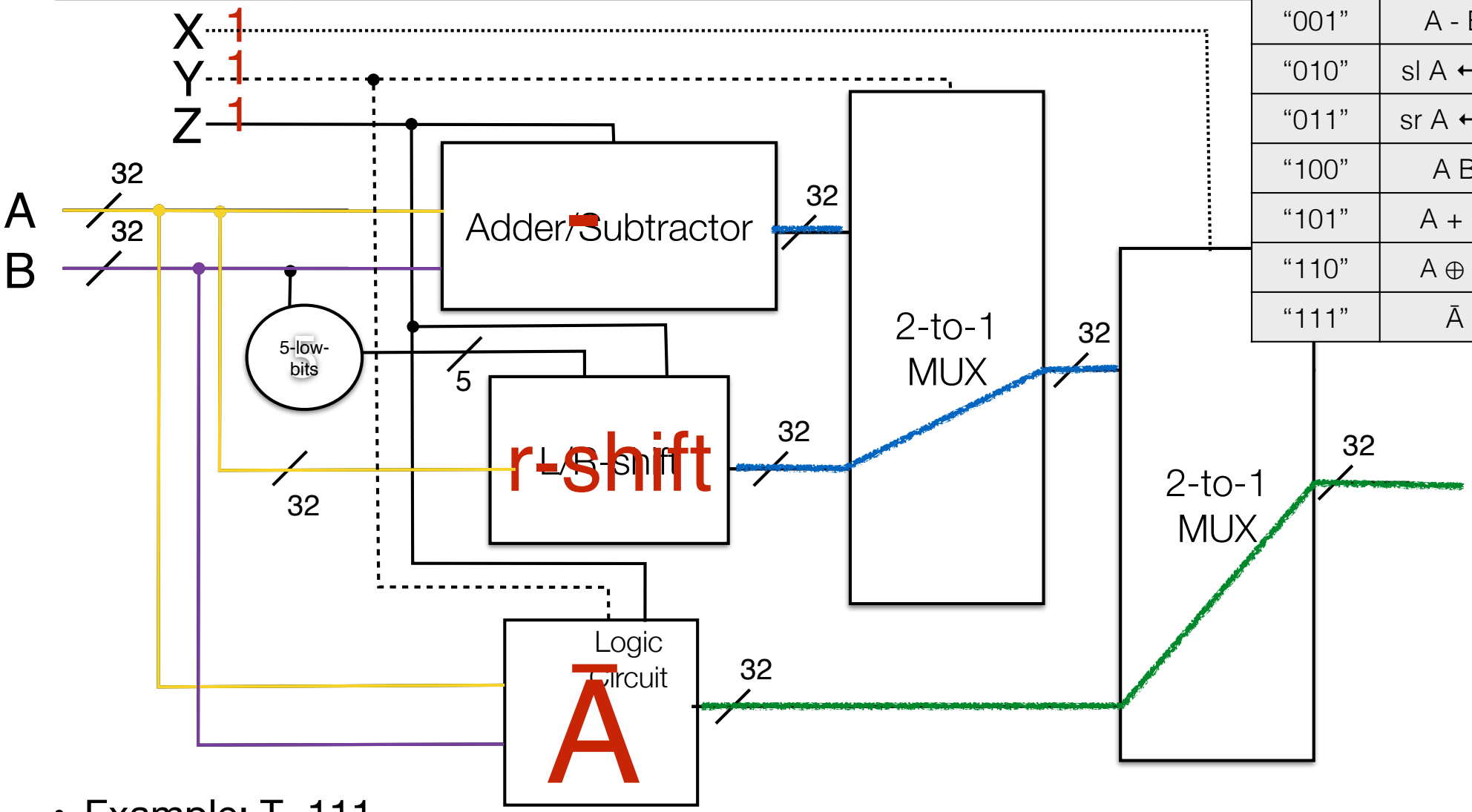
T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā



- Example: T=110

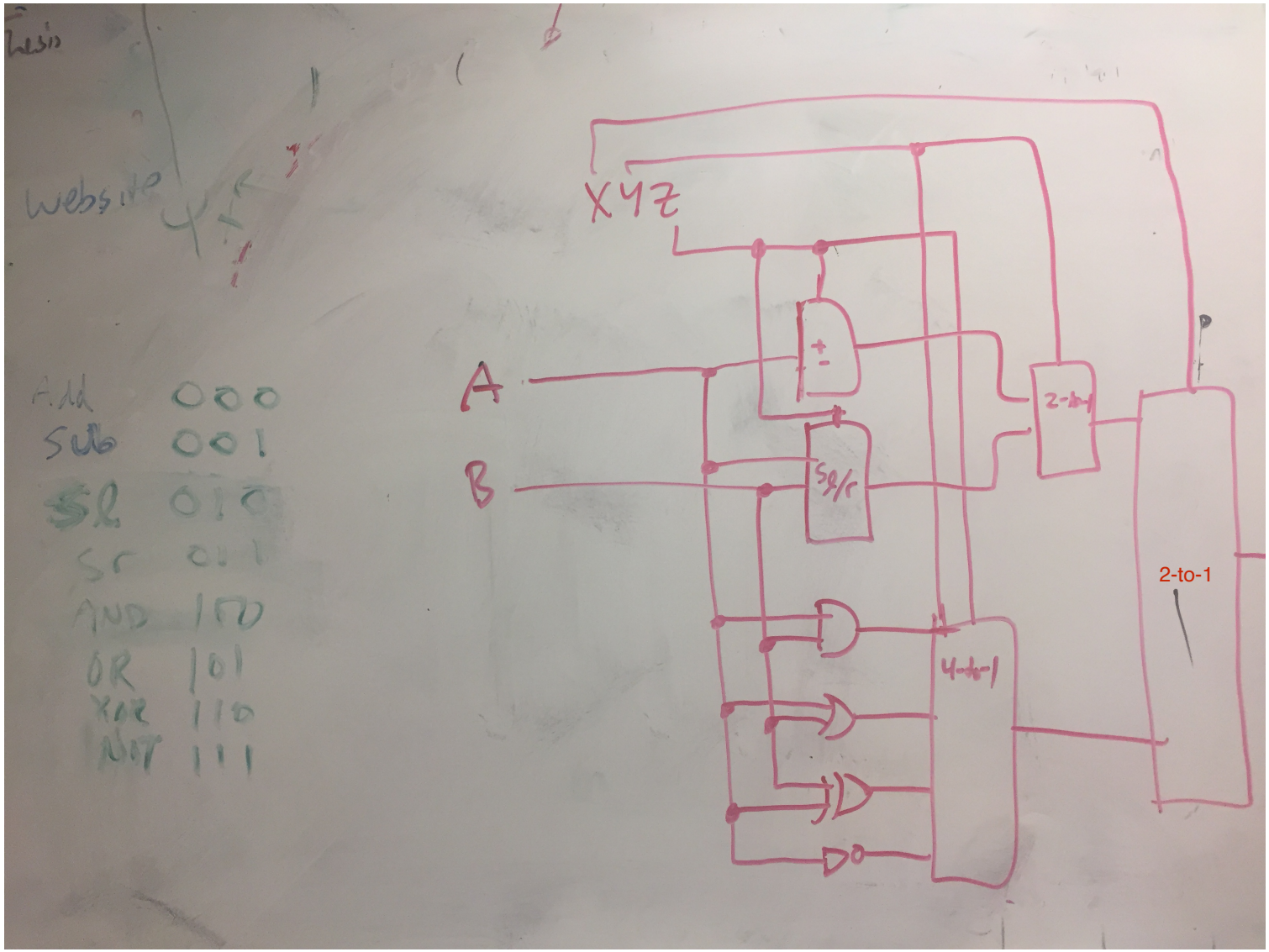
Putting it all together

T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā



- Example: T=111

Prof. Rubenstein sketching this out…



T=XYZ	Output
“000”	A + B
“001”	A - B
“010”	sl A ← B
“011”	sr A ← B
“100”	A B
“101”	A + B
“110”	A ⊕ B
“111”	Ā