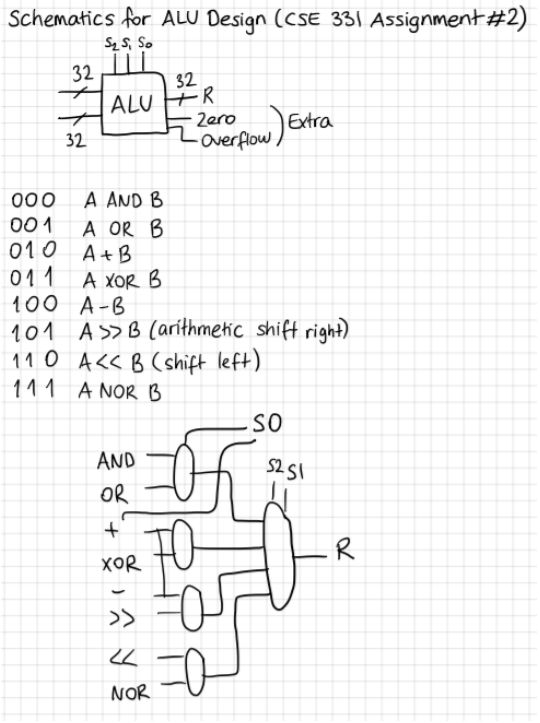
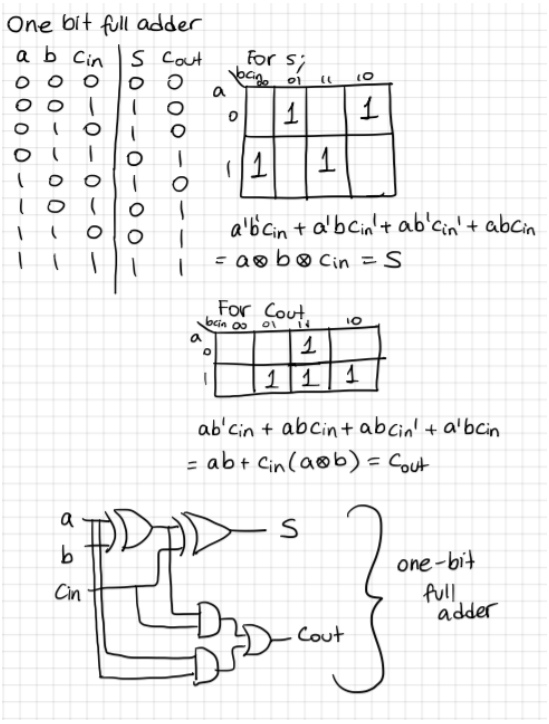
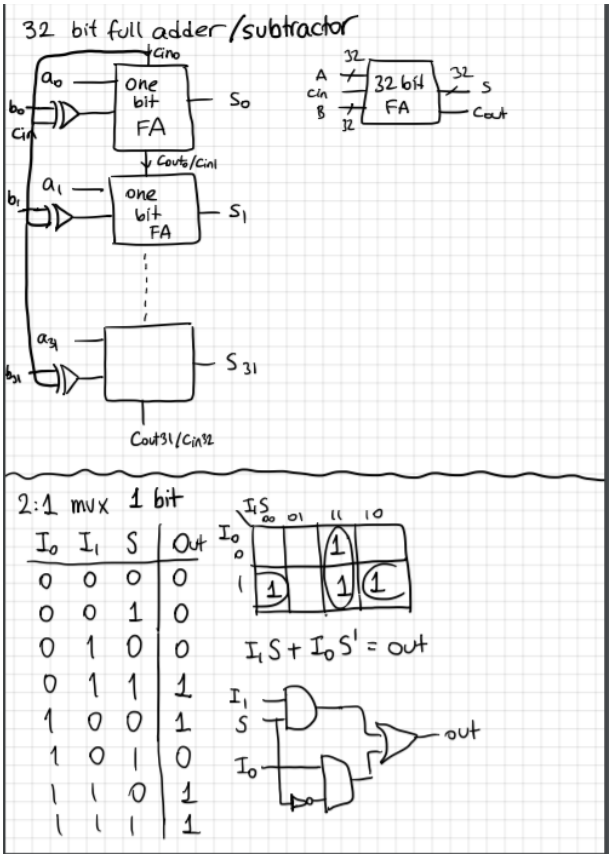
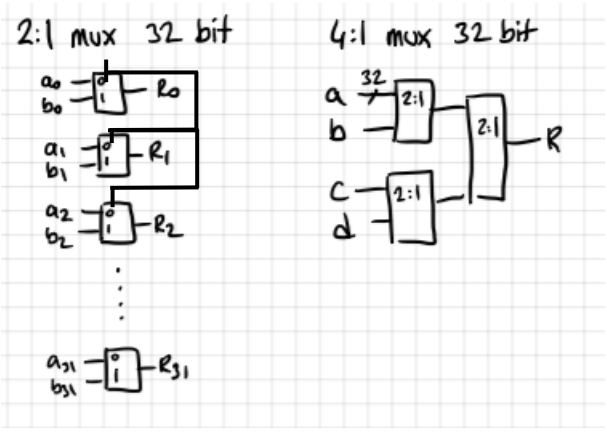
CSE331 Computer Organization Assignment #2 Report - Gökçe Nur Erer 171044079

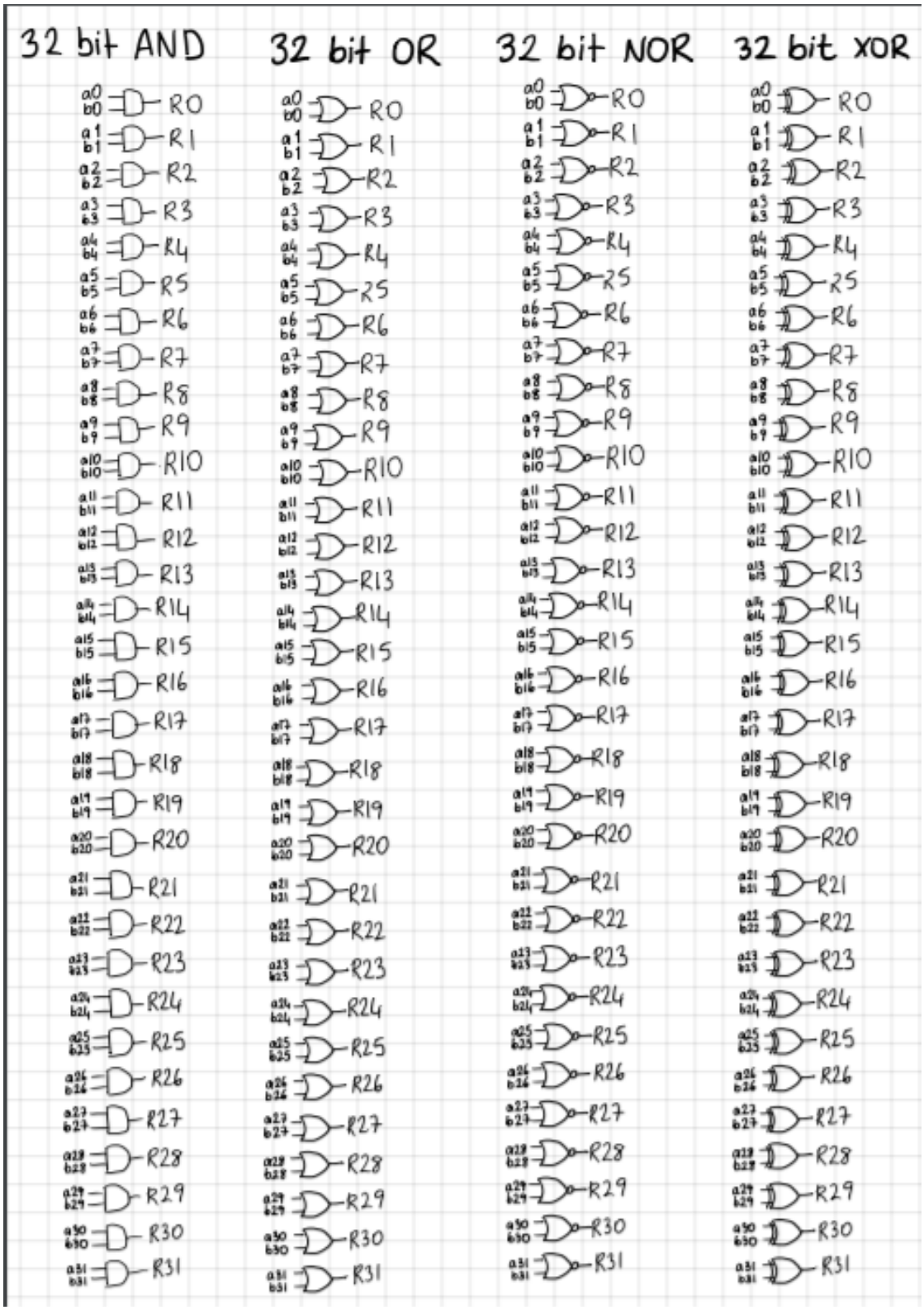
ALU Design Schematics

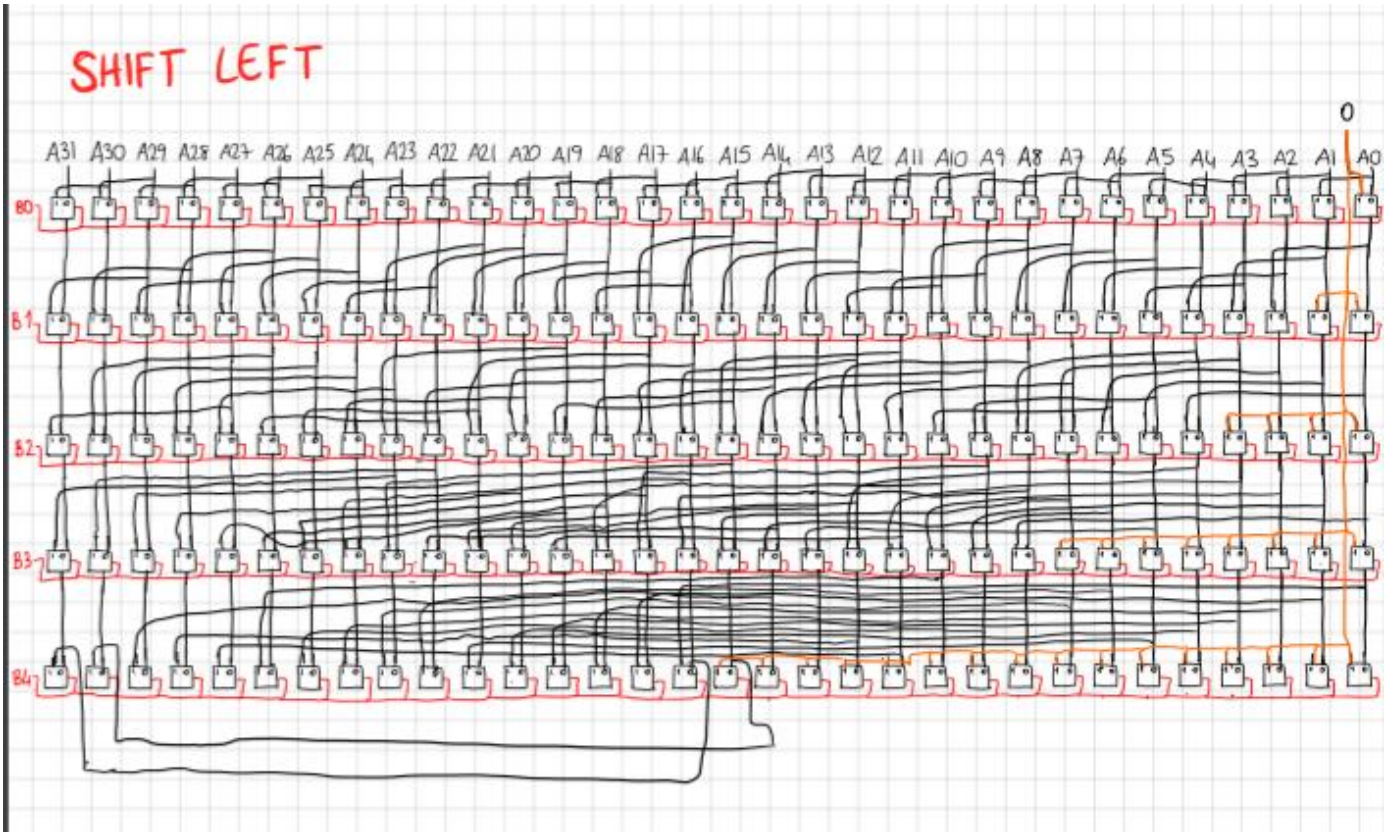


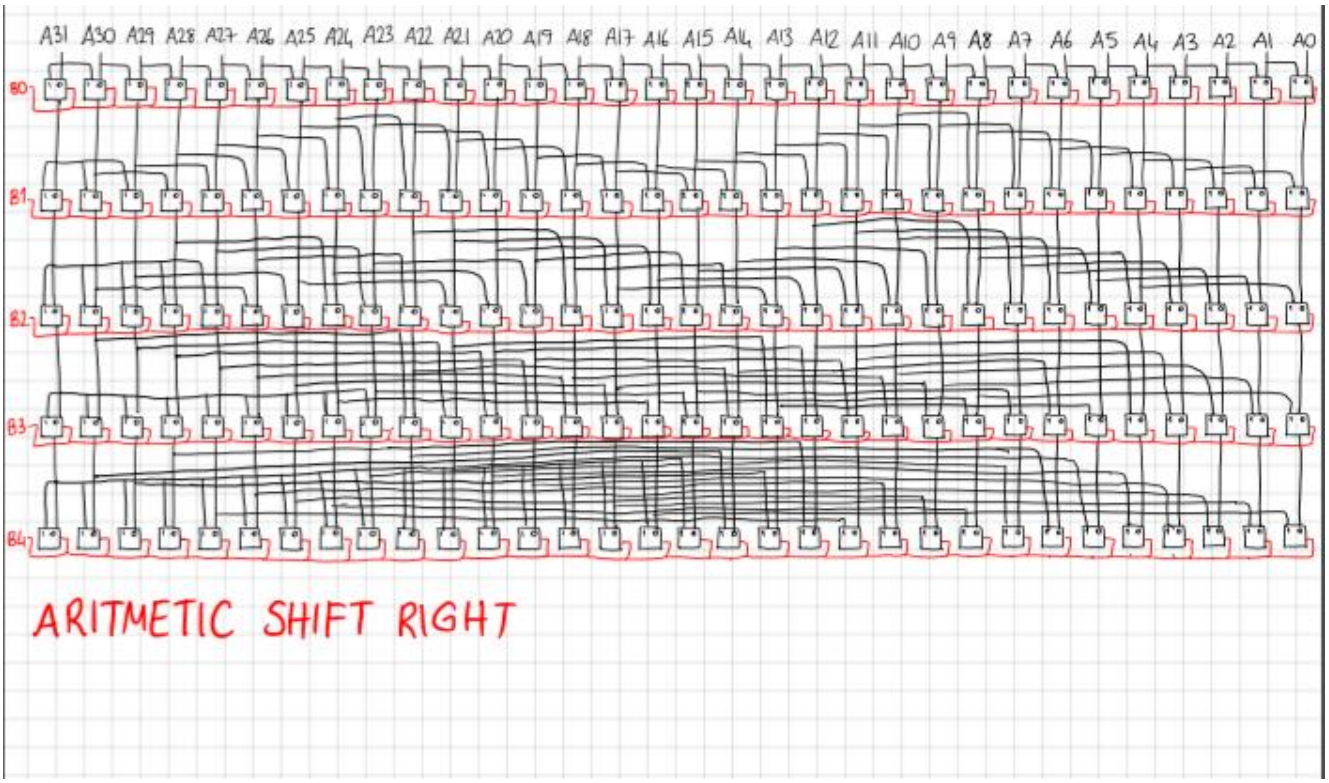







32 bit AND
32 bit OR
32 bit NOR
32 bit XOR
a0 b0 R0
a1 b1 R1
a2 b2 R2
a3 b3 R3
a4 b4 R4
a5 b5 R5
a6 b6 R6
a7 b7 R7
a8 b8 R8
a9 b9 R9
a10 b10 R10
a11 b11 R11
a12 b12 R12
a13 b13 R13
a14 b14 R14
a15 b15 R15
a16 b16 R16
a17 b17 R17
a18 b18 R18
a19 b19 R19
a20 b20 R20
a21 b21 R21
a22 b22 R22
a23 b23 R23
a24 b24 R24
a25 b25 R25
a26 b26 R26
a27 b27 R27
a28 b28 R28
a29 b29 R29
a30 b30 R30
a31 b31 R31


SHIFT LEFT
0
A31 A30 A29 A28 A27 A26 A25 A24 A23 A22 A21 A20 A19 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
B0
B1
B2
B3
B4


A31 A30 A29 A28 A27 A26 A25 A24 A23 A22 A21 A20 A19 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
B0
B1
B2
B3
B4
ARITMETIC SHIFT RIGHT

Verilog Modules And Their Descriptions

one_bit_full_adder

This module is the module which adds 2 1-bit binary numbers by using xor, and & or gates.

full_adder_32_bit

This module is the module which adds or subtracts 2, 32-bit binary numbers by using one bit full adders. To give the subtraction signal, this adder uses the select signal's most significant bit (s[2]) and uses an xor gate to convert the second number if necessary.

and32bit/or32bit/xor32bit/or32bit

These modules do the basic logical operations for 32-bit 2 binary numbers. In each of these modules their 1 bit versions are used to do the operation for each single digit.

mux_2_1_for_1_bit

These module creates a 2:1 mux which takes 1 bit inputs by using 2 and gates, one or gate and an inverter.

mux_2_1_for_32_bit

These module uses 2:1 1-bit muxes to create a 2:1 32-bit mux. Each digit of each number is processed by a single 2:1 1-bit mux, which makes the total mux count 32. If the select bit of all the muxes are 0 the first number is chosen, if the select bit of all the muxes are 1 then the second number is chosen.

mux_4_1_for_32_bit

These module uses 2:1 32-bit muxes to create a 4:1 32-bit mux. Each two of the four input numbers are processed by a single 2:1 32-bit mux, which makes the total mux count 2 for the first layer. From those 2 muxes one output has to be selected so another 2:1 32-bit mux decides which one to choose.

For this whole 4:1 32-bit mux, 2 select bits are needed. For the first layer of muxes the least significant bit of the select input enters to both muxes. For the second layer of muxes the most significant bit of the select input enters to the mux.

shifter0to1/shifter0to2/shifter0to4/shifter0to8/shifter0to16

These modules are for the arithmetic shift right module's layers. The whole connections can be seen in the schematic above. First layer → shifter0to1 Second Layer→shifter0to2..etc

ArithmeticShiftRight32bit

This module uses shifter0to1/shifter0to2/shifter0to4/shifter0to8/shifter0to16 modules to use as its layers of shifting and results a 32 bit shifted number. This module shifts a 32 bit number aritmetically right by another given 32 bit number. The whole logic behind this module can be seen in the schematic above.

shifterLeft0to1/shifterLeft0to2/shifterLeft0to4/shifterLeft0to8/shifterLeft0to16

These modules are for the shift left module's layers. The whole connections can be seen in the schematic above. First layer → shifterLeft0to1 Second Layer→shifterLeft0to2..etc

shifterLeft32bit

This module uses shifterLeft0to1/shifterLeft0to2/shifterLeft0to4/shifterLeft0to8/shifterLeft0to16 modules to use as its layers of shifting and results a 32 bit shifted number. This module shifts a 32 bit number logically left by another given 32 bit number. The whole logic behind this module can be seen in the schematic above.

Alu32

This module is the top module where the whole ALU implementation is. This module calculates AND,OR,+,XOR,-,>>,<<,NOR operations and puts them as input into a layer of muxes. Each 2 operation enters a 2:1 32-bit mux first, then the resulting outputs enter into a 4:1 32-bit mux which gives us the overall result.

For the first layer of muxes the select bit is s0, s0 is also connected to the adder's cin input.

For the 4:1 mux the select bits are s2 and s1.

Simulation Result