CS251, Spring 2022
Assignment 3: Question 1

Q1a) Perform the unsigned multiplication of 1010 (multiplicand) x 1101 (multiplier) by completing the tables below. For those of you who took more than one method, use the first multiplier method you took in lecture. You may not need all the rows in the table.

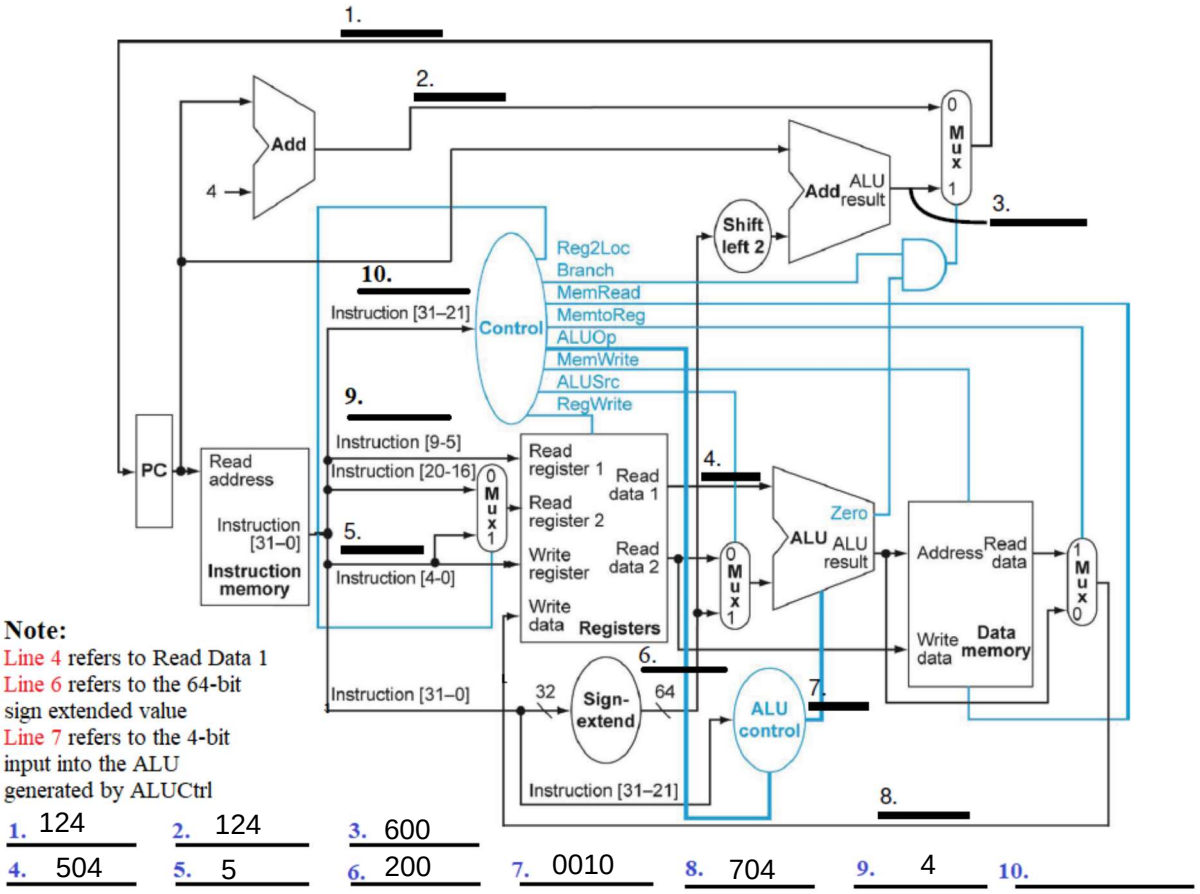
Interaction	Multiplier	Multiplicand	Product
Initialize	1101	0000 1010	0000 0000
1	0110	0001 0100	0000 1010
2	0011	0010 1000	0000 1010
3	0001	0101 0000	0011 0010
4	0000	1010 0000	1000 0010

2. (6 points) Floating Point

   Write the base 10 number -71.09375×10^{-2} as a 32-bit, IEEE normalized floating point number with biased exponent. First you must *simplify the scientific notation* and then begin the conversion to binary. Follow the algorithm discussed in class to convert the fractional binary portion of the number to binary.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	0	1	1	1	1	1	1	0	0	1	1	0	1	1	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

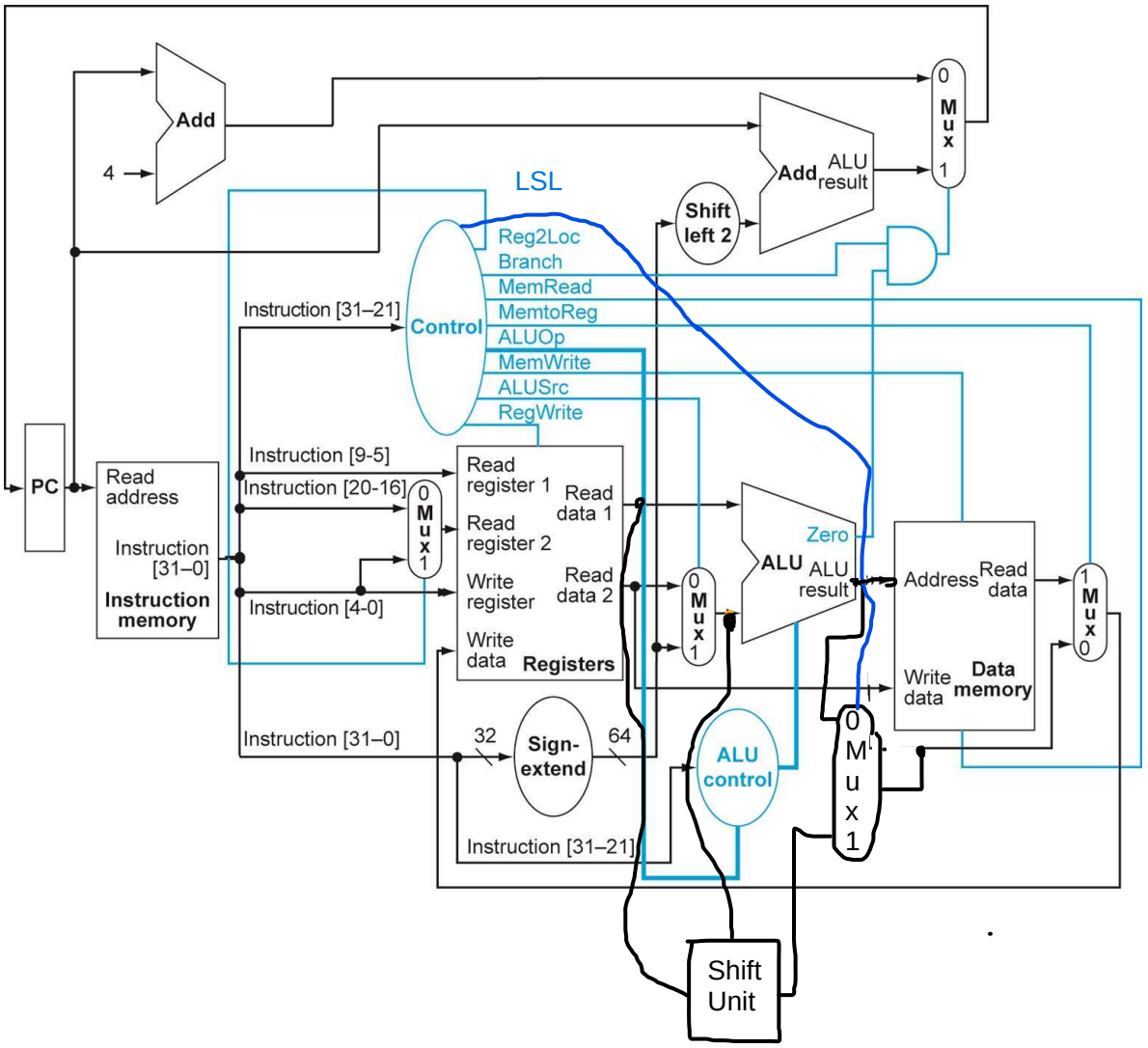


1. 124 2. 124 3. 600
4. 504 5. 5 6. 200 7. 0010 8. 704 9. 4 10.

(b) (9 points)

The `ADDI` instruction **does not** require any physical hardware changes to the single cycle datapath. Fill in the table below indicating the value of all existing control lines necessary to execute the `ADDI` instruction on the single cycle datapath. You **must use don't cares** where appropriate.

Type	Reg2 Loc	ALU Src	Mem ToReg	Reg Write	Mem Read	Mem Write	Brch	ALU op1	ALU op0
R-format	0	0	0	1	0	0	0	1	0
`LDUR`	X	1	1	1	1	0	0	0	0
`STUR`	1	1	X	0	0	1	0	0	0
`CBZ`	1	0	X	0	0	0	1	0	1
`ADDI`	X	1	0	1	1	0	0	0	0


Add
4
LSL
Shift left 2
Add
ALU result
M u x
Reg2Loc
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–21]
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [9-5]
Instruction [20-16]
Instruction [4-0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
Instruction [31–0]
32
Sign-extend
64
ALU control
Instruction [31–21]
Shift Unit

(b) (4 points) In the table below, give the settings of the control bits to implement the new `LSL` ARM instruction. Use Don't Cares where appropriate. If you need an extra control line to implement this instruction or if you need to increase the number of bits in a control line, add additional columns to the table for the new control line, split a column to increase the number of bits in a control line, and in either case include a note below explaining the effect of the new/increased control line(s) on the datapath and what its setting should be for other instructions. Make sure you do not break any other instructions. You should be able to determine the purpose and effects of each of the control signals from the datapath diagram on the previous page.

Type	Reg2 Loc	ALU Src	Mem ToReg	Reg Write	Mem Read	Mem Write	Brch	ALU Op1	ALU Op0	LSL
LSL	X	1	0	1	0	0	0	X	X	1

State the value of any new control signal(s) for all other ARM instructions:

5. (5 points) Datapath Timing

Consider the Single-cycle processor shown in the previous question. Suppose the circuit elements take the following times: Instruction memory: 200ps, Register read: 50ps, Register write: 50ps, ALU and all adders: 200ps, Data memory: 200ps, Main Control Unit: 15ps, ALU Ctrl 10ps, Sign Extend & Shift Left Units: 5ps each.
Assume that PC and MUXes do not take any time.

Compute the execution time for each instruction type below:

R-format:	525	ps.	Instruction Fetch + Register Read + Register write + ALU + Main Control + ALU Control
LDUR:	715	ps.	Instruction Fetch + Register Read + Register Write + ALU + Data Access + ALU Control + Sign Extend
STUR:	665	ps.	Instruction Fetch + Register Read + ALU + Data Access + ALU Control + Sign Extend
CBZ:	470	ps.	Instruction Fetch + Register Read + ALU + ALU Control + Sign Extend + Shift Left
B:	260	ps.	Instruction Fetch + Register Read + Sign Extend + Shift Left