Microarchitecture Design

Spartak Gevorgyan, Vladislav Semenyutin,
Shohin Abdulhamidov

ARM Instructions and Assembly Program

Single Data Transfer

Assembly Language Symbol	Instruction
LDR	Load register from memory
STR	Store register to memory

Branch

Assembly Language Symbol	Instruction
BLT	Branch if less than

Data Processing

Assembly Language Symbol	Instruction	OpCode
MOV	Move register or constant	1101
ADD	Addition of operands from registers	0000
CMP	Compare values in register	1010
SUB	Subtraction of operands from registers	0010
ORR	Bitwise OR operation	1100
ADC	Addition of operands with carry	0101

Assembly Program

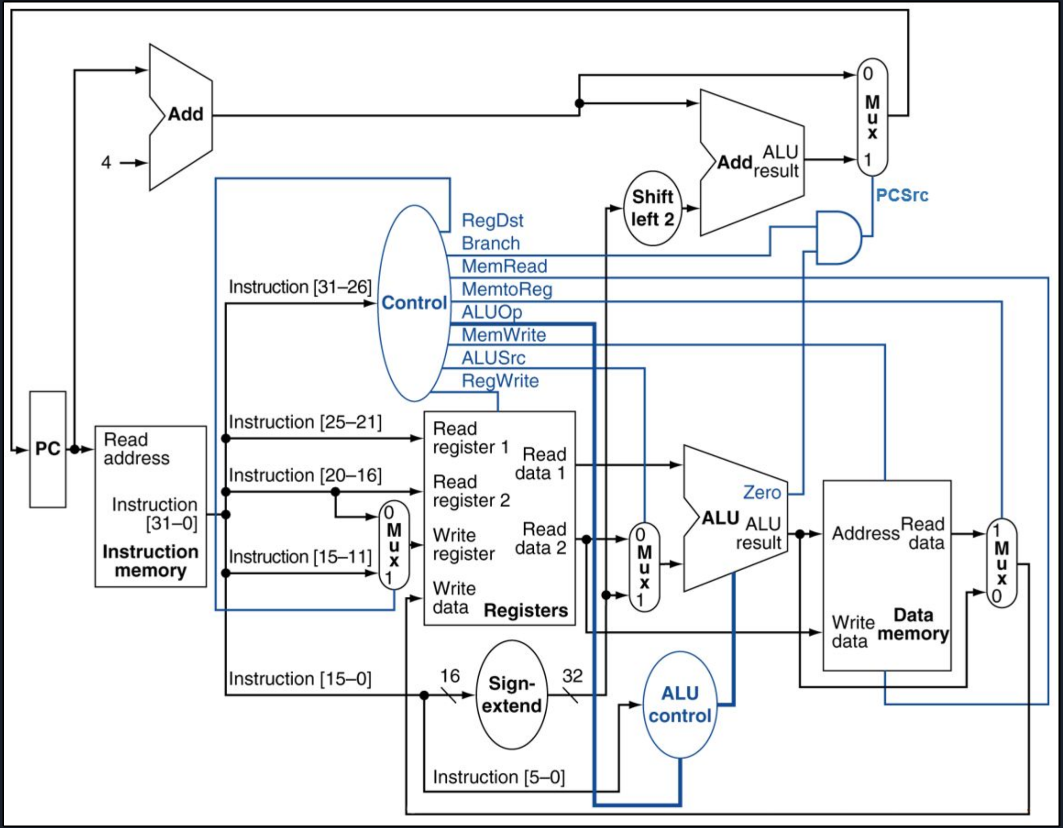
```
MOV R4, #0          ; register to store the sum
MOV R0, #0          ; index i, will be used in the loop
MOV R3, #16         ; final size of the array, 4 words
LDR R1, =A          ; load array A into register 1
LDR R2, =B          ; load array B into register 2
LDR R7, =C          ; load array C into register 7, it will be the empty array where results will be saved
LOOP:
    LDR R5, [R1], #4  ; load value of array A into register 5 and update base register by R1 = R1 + 4
    LDR R6, [R2], #4  ; load value of array A into register 6 and update base register by R2 = R2 + 4
    ADD R4, R5, R6    ; add values from register 5 and 6 and place the result in register 4
    STR R4, [R7], #4  ; store value into R7 and update base register by R7 = R7 + 4
    ADD R0, #4        ; increase counter by 1 word = 4 bytes
    CMP R0, R3        ; check if we are done
    BLT LOOP          ; if not done loop again
```

Instructions

We can divide the instructions into 3 categories

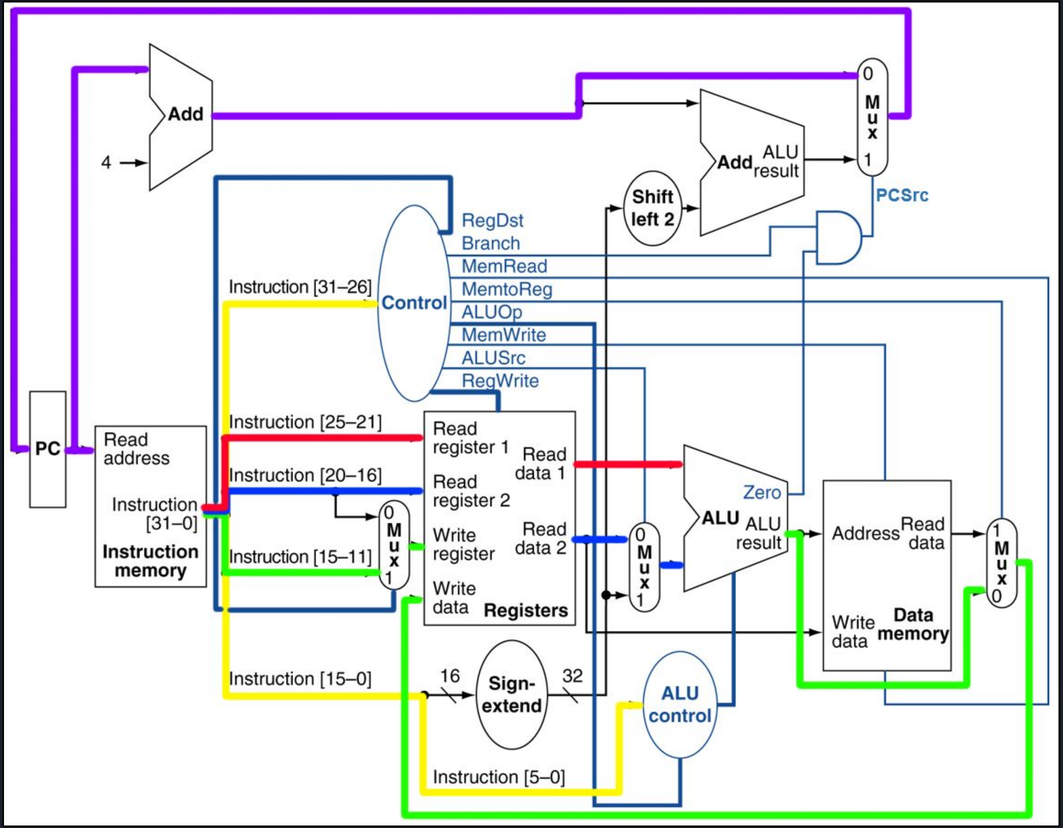
Add/Sub (R-type)	0	rs	rt	rd	shamt	funct
	31:26	**25:21**	**20:16**	**15:11**	**10:6**	**5:0**
Load/Store (I-type)	35 or 43			rs	rt	off 16 (signed 16 bit)
	31:26	**25:21**	**20:16**	**15:0**		
Branch (I-type)	4			rs	rt	off 16 (signed 16 bit)
	31:26	**25:21**	**20:16**	**15:0**		

Data Path



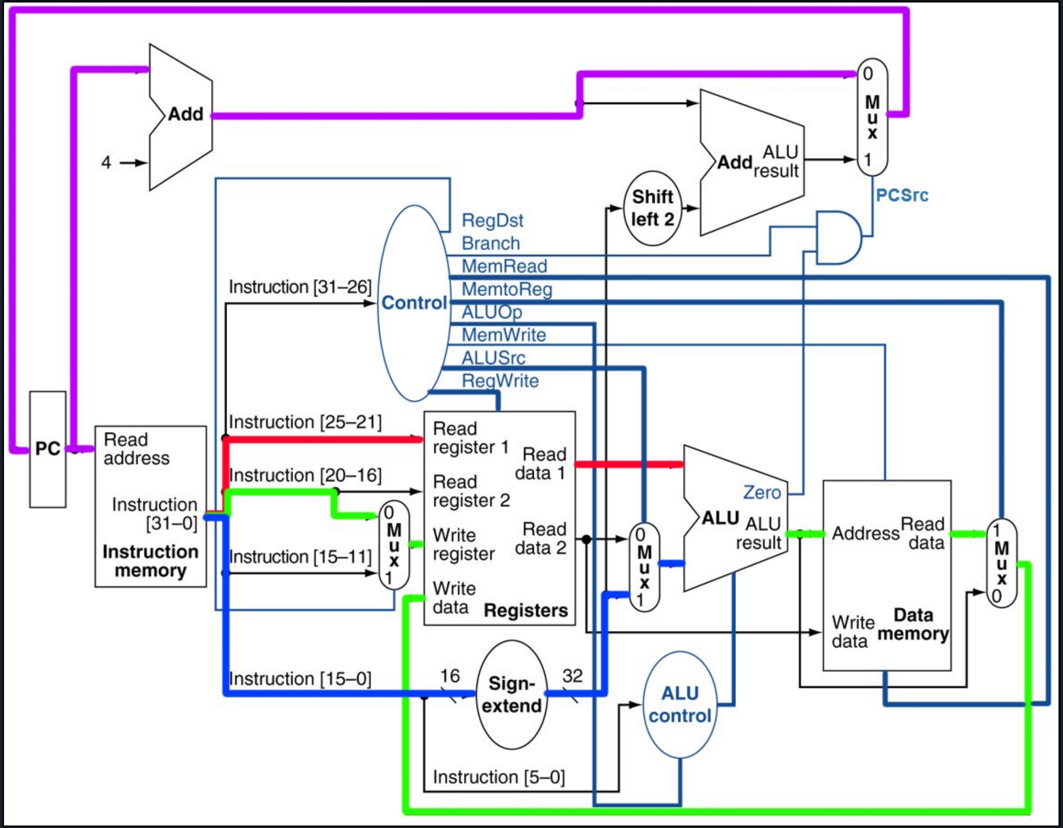
Add
4
Add
ALU
result
Shift
left 2
0
M
u
x
1
PCSrc
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
PC
Read
address
Instruction
[31–0]
Instruction
memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
0
M
u
x
1
Read
register 1
Read
register 2
Write
register
Write
data
Read
data 1
Read
data 2
Registers
0
M
u
x
1
Zero
ALU
ALU
result
Address
Read
data
Write
data
Data
memory
1
M
u
x
0
Instruction [15–0]
16
Sign-
extend
32
ALU
control
Instruction [5–0]

R-type

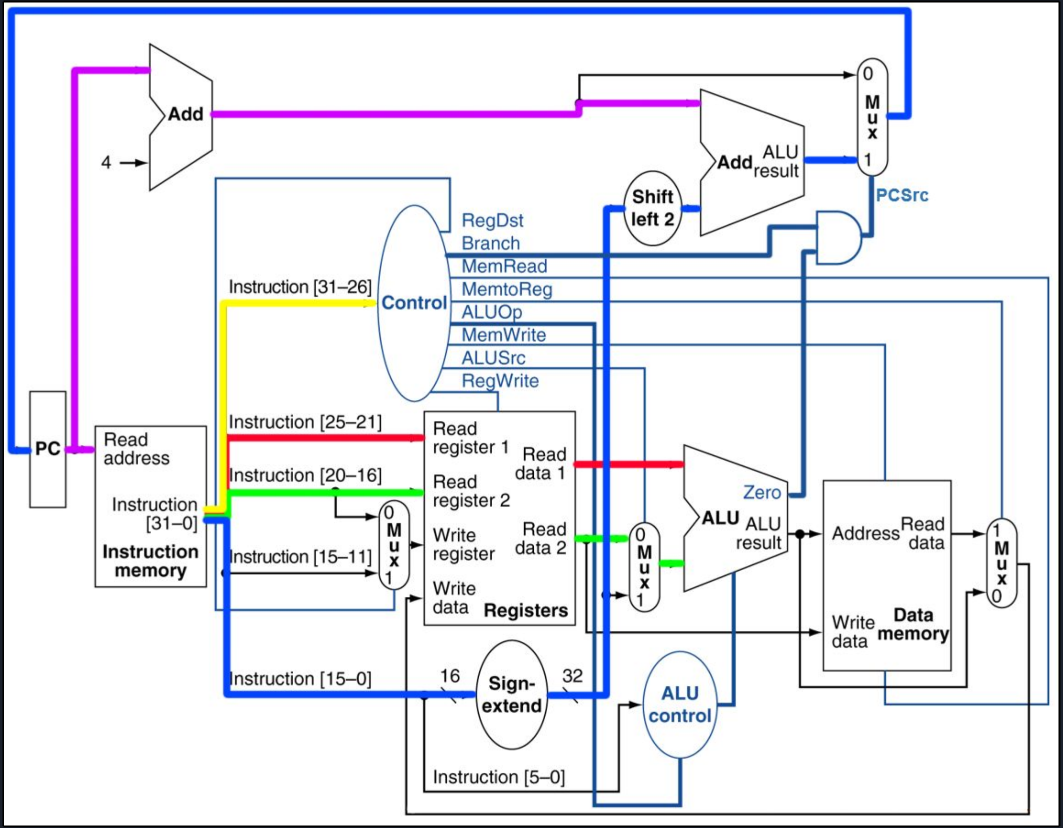


Add
4
Add
ALU
result
0
M
u
x
1
PCSrc
Shift
left 2
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
PC
Read
address
Instruction
[31–0]
Instruction
memory
Read
register 1
Read
register 2
Write
register
Write
data
Read
data 1
Read
data 2
Registers
0
M
u
x
1
Zero
ALU
ALU
result
Address
Read
data
Write
data
Data
memory
1
M
u
x
0
16
Sign-
extend
32
ALU
control

Load/Store (I-type)



Branch (I-type)



Truth Table

Signal name	**R-format**	**lw**	**sw**	**beq**
RegDst	1	0	X	X
ALUSrc	0	1	1	0
MemtoReg	0	1	X	X
RegWrite	1	1	0	0
MemRead	0	1	0	0
MemWrite	0	0	1	0
Branch	0	0	0	1
ALUOp1	0	0	0	1
ALUOp0	0	0	0	1