Problem 2

Given the hardware ingredients ALU, register file (R), memories etc. on the next page. Design datapath, and control unit table of a 16 bits processor that can handle the instructions:

Instruction	Action	Instruction format
JACL ra, imm	if AC < imm then PC ←(R[ra])[7:0]	Opcode: 1 1 1 1 1 | ra: | imm:
SUBd ra, dir	AC←R[ra] - M(dir)	Opcode: 1 1 1 1 0 | ra: | dir:
ORd ra, dir	M(dir)←R[ra] OR AC	Opcode: 1 1 1 0 0 | ra: | dir:

You may use additional hardware elements like registers, multiplexors etc. The memories are addressed on word level. Take care of:

- Minimising the number of elements; only the given instructions have to be handled.
- Give clearly the bus widths; not all elements have the same number of bits

Description of the provided hardware elements

Registers (PC, AC):

- The register content is constantly in gate OUT[:]
- LOAD = 1 → IN[:] is stored in the register on rising clock edge

Register file :

- The content of register D[2:0] is continuously in gate OUT[15:0]
- Write = 1 → IN[15:0] is stored in register D[2:0] on rising clock edge

Instruction memory:

- The content of position DIR[7:0] is constantly in gate Dataout[15:0]

Data memory:

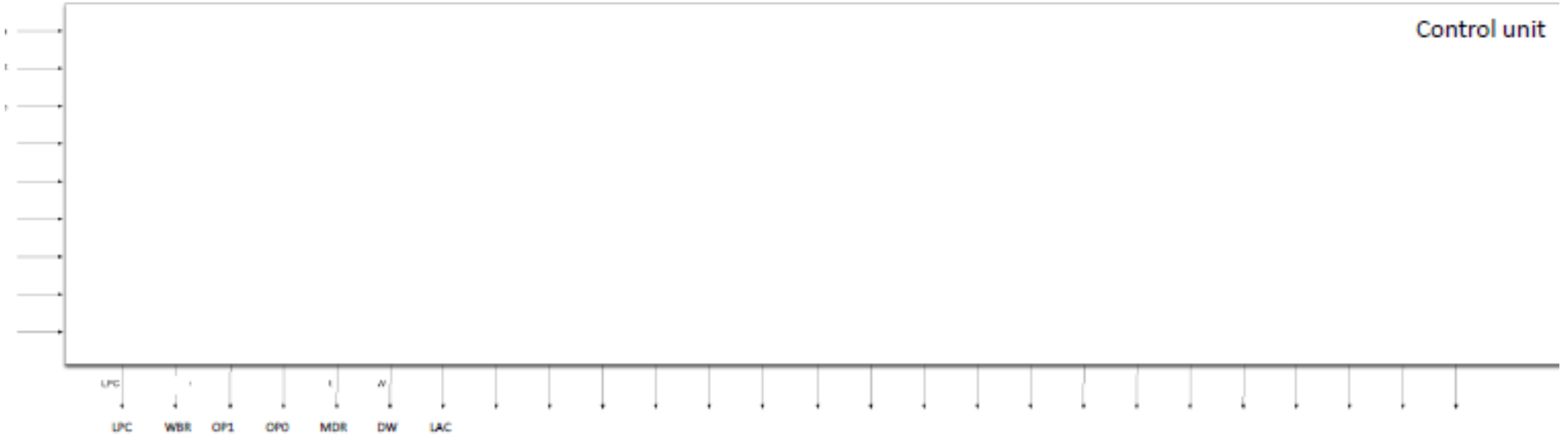
- Read = 1 → The content of position DIR[7:0] appears at gate Dataout[15:0]
- Write = 1 → Datain[15:0] is stored into memory position DIR[7:0] on rising clock edge
- If Read = 0 and Write = 0 → Dataout[15:0] is disconnected, no signal

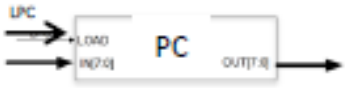
ALU:

- OP[1:0] = 00 → OUT[15:0] = A[15:0] + B[15:0] (arithmetic sum)
- OP[1:0] = 01 → OUT[15:0] = A[15:0] - B[15:0] (arithmetic subtract)
- OP[1:0] = 10 → OUT[15:0] = A[15:0] AND B[15:0] (bitwise AND)
- OP[1:0] = 11 → OUT[15:0] = A[15:0] OR B[15:0] (bitwise OR)
- N is enabled (=1) if OUT[15:0] is a negative number
- Z is enabled (=1) if OUT[15:0] is zero
- C is enabled (=1) if a carry out occurs from the operation performed by the ALU

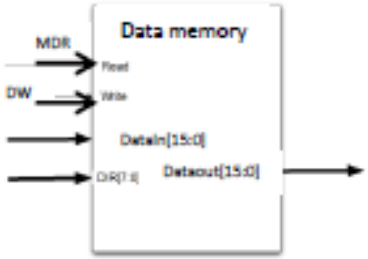
Control Table

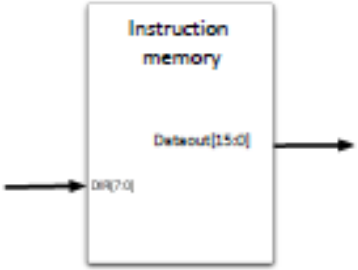
	Control signals													
Instruction														
JACL														
SUBd														
ORd														


Control unit
LPC
WBR
OP1
OP0
MDR
DW
LAC


LPC
LOAD
IN[7:0]
PC
OUT[7:0]


WBR
Write
D[2:0]
IN[15:0]
Register file
OUT[15:0]


MDR
DW
Reset
Write
Datain[15:0]
DIR[7:0]
Data memory
Dataout[15:0]


Instruction memory
Dataout[15:0]
DIR[7:0]


LAC
LOAD
AC
IN[15:0]
OUT[15:0]


OP
A[15:0]
B[15:0]
ALU
N,Z,C
OUT[15:0]
n, z, c

Data path