1.2 SOFTWARE & SOFTWARE DEVELOPMENT · 1.2.4 · FULL A-LEVEL

Assembly, the LMC & addressing modes

The LMC instruction set, tracing and writing programs, and the **four addressing modes**. Spec 1.2.4(c)(d).

01 LMC instructions

INP / OUT input to ACC / output ACC.

LDA / STA load from / store to an address.

ADD / SUB add/subtract value at address to/from ACC.

BRA / BRZ / BRP branch always / if zero / if ≥ 0.

HLT / DAT halt / reserve a data location.

Registers Loads & arithmetic change the **ACC**; branches change the **PC**.

02 Assembly vs high-level

Assembly

~1:1 with machine code. Fast, small, direct hardware control. Hard to write, not portable.

High-level

One line = many instructions. Easier, quicker, portable; needs translation, less control.

Translator Assembly → machine code via an **assembler**.

03 Four addressing modes

Immediate Operand **is the value** (no memory access).

Direct Operand is the **address of the value** (LMC's normal mode).

Indirect Operand is the **address of the address** — reaches a bigger address space.

Indexed Real address = operand + **index register** — step through arrays.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Seven slips on assembly and addressing questions.

01

Trace LMC **on paper**, writing the ACC after every line. Never in your head.

02

BRZ = branch on zero only; **BRP** = branch on positive **or zero**.

03

Every named location needs a **DAT** line. `x DAT 0` reserves x holding 0.

04

"Changes the ACC" → LDA/INP/ADD/SUB. "Changes the PC" → BRA/BRZ/BRP.

05

Modes: immediate = **value**; direct = **address**; indirect = **address of address**; indexed = **+ index register**.

06

Indexed addressing is the **array** mode — increment the index register to move along.

07

Assembly vs high-level "discuss": speed/size/control vs development time/readability/portability — and conclude for the scenario.