

Unassessed Course Work 4

1. (i) Write a TOY1 assembler program (instructions plus constants) to change the sign of all negative numbers in a list of 32 integers stored consecutively from memory address 1F0H upwards.

Hint: Develop a high-level language solution first.

- (ii) Translate your program into binary TOY1 machine instructions, indicating clearly the address of each instruction/constant.

2. TOY2 is a 32-bit successor to TOY1. Its instructions include:

$\text{Register}[N] = \text{Memory}[\text{Address}]$

$\text{Memory}[\text{Address}] = \text{Register}[N]$

$\text{Register}[N] = \text{Register}[M]$

$\text{Register}[N] = \text{Register}[M] + \text{Register}[P]$

$\text{Register}[N] = \text{Register}[M] - \text{Register}[P]$

Devise a 32-bit instruction format or formats for these instructions. For your format(s) state

- (i) the number of instructions catered for
- (ii) the number of registers catered for
- (iii) the number of words in memory catered for

Note: There is no single correct format; have a go at designing your own formats!