Machine cycle drama script

The players

- Input
- Control
- RAM
- Address register
- Program counter
- Instruction register
- ALU
- Accumulator
- Interpreter

Script

INPUT

I think something's coming my way. It looks like a program.

CONTROL

Will we be busy with this one?

INPUT

No, it's just another of those four liners the beginners use once a year in that programming course. And they call this programming! Should we treat it seriously?

CONTROL

Of course. As always. *AL* doesn't know how to make judgements anyway.

INPUT

Here it comes:
A=3

B=4

C=A + B

PRINT C

That's all.

CONTROL

Send the whole lot to RAM

DIGITAL TECHNOLOGIES HUB

RAM

OK. I'll store it and I'll give each line an address so we can keep track of where we're up to.

address 1036 A=3

address 1037 B=4

address 1038 C=A + B

address 1039 print C

How's that?

CONTROL

It looks like we're ready to start on it. Now, registers, get ready ...

Address register: you'll be minding the four addresses in RAM of each of the lines of the program.

Program counter register: you'll only have to keep the address in RAM of the current line of the program.

Instruction register: your job is important. You'll only have to mind one line of the program from RAM but it will be the one we're doing at that moment.

Accumulator: I realise you think you're special because you're storing the answers that *ALU* gets for us, but keep in mind you're just a register too. You can take a break – you won't be busy for at least a few billionths of a second.

RAM: send all information on to the registers.

RAM

Done!

ADDRESS REGISTER

I've got the four addresses of the program lines, but that's all: 1036, 1037, 1038, 1039. RAM's now got the actual contents at those addresses.

PROGRAM COUNTER REGISTER

I'm pointing to the first one: 1036

INSTRUCTION REGISTER

I see 'A=3' sitting in RAM at that address and I've copied it. But I have no idea what it means.

CONTROL

Interpreter, would you kindly translate that into *machine code* and we'll store it in a *data register* at a location we will call 'A'? Thanks.

Now, *program register*: Increment your address counter please.

PROGRAM COUNTER REGISTER

OK. I'm now pointing to address 1037

INSTRUCTION REGISTER

I see ‘B=4’ at that address and I've copied it. But I don't understand it.

CONTROL

Interpreter, would you kindly translate that into *machine code* and we'll store that in a data register called ‘B’? Thanks.

Let's count through one machine cycle.

Now, *program register*: increment your address counter.

PROGRAM COUNTER REGISTER

OK. It's pointing to address 1038 in RAM.

INSTRUCTION REGISTER

I have a line reading ‘C = A + B’ at that address in RAM. But what does that mean?

CONTROL

OK. This is where it really happens! Everyone ready?

First we send those earlier numbers from the data registers along to the *ALU*.

Interpreter, would you kindly translate ‘A + B’ into *machine code* for me so I can use my *microprograms* on it?

Data register, please send both those numbers ‘A’ and ‘B’ you are keeping to *ALU.* That ends our instruction/fetch phase. Now for our *execution phase.*

So *AL*, please do that calculation for us. One addition shouldn't strain you too much after practising on millions for that spreadsheet application yesterday.

ALU

Done!

CONTROL

Good, *AL*, send that result to the *accumulator register.*

ACCUMULATOR

Got it!

[Editor's note: ALU performs addition]

CONTROL

Now *program register*: please add 1 to your address counter to end this execution phase. I hope whoever's out there realises this machine cycle takes up quite a few *clock cycles* to get through. It's not as simple as it might look! I have to keep using so many microprograms to deal with each instruction in the program – even running at 3 Gigahertz!

PROGRAM COUNTER REGISTER

Are you all finished? In case you've forgotten I'm now pointing to address 1039

CONTROL

Instruction register, what does that address in RAM have?

INSTRUCTION REGISTER

It's 'print C'. What does that mean?

CONTROL

Interpreter, would you kindly translate that 'print' instruction into *machine code* so I can use my *microprograms* I've got stored permanently in *ROM* to do it?

INTERPRETER

Done!

CONTROL

That's better! *Accumulator*, please send the result along the *bus* to whichever printer or other output device is designated by the *system software program* settings held in RAM. Now what do each of you registers have stored?

ADDRESS REGISTER

I've still got the four RAM addresses of each of the lines of the program.

PROGRAM COUNTER REGISTER

I'm still pointing to that last address 1039.

INSTRUCTION REGISTER

I'm still storing the last line of the program: 'print C'.

ACCUMULATOR

I sent that final answer but I'm still storing it (just in case it's lost). Is anyone interested in hearing it is 7?

CONTROL

Not really. Doesn't mean anything to me.

Thanks guys. Few will ever realise how much work it takes us just to add two numbers. They think it's so easy for us. Let's see – 20 nanoseconds, that's 20 billionths of a second. Time to take a break.