

## COS-10004 (Computer Systems)

**Name:** Dave Nguyen (Nguyen Quang Anh).

**ID:** 104697710.

Q10.1:

The screenshot displays a computer system simulator interface. On the left, the 'Program' window shows assembly code for a pixel-drawing routine. The code includes a 'flash' section and a 'drawpixel' function that uses indirect addressing with registers R1, R2, and R3. A comment indicates that R3 is a local variable. The 'delay' function uses a timer to control the drawing process. On the right, the 'Processor' window shows the current state of the system, including the Program Counter (PC), registers (R0-R15), and status bits (NZCV). The 'Input/Output' window at the bottom shows the current state of the system, including the 'Count' and 'Current' registers.

```
1|flash:
2|  MOV R0, #.green
3|  BL drawpixel
4|  MOV R0, #.white
5|  BL drawpixel
6|  B flash
7|  HALT
8|drawpixel:
9|  PUSH {LR,R1,R2} //R3 is basically local variable
10| MOV R1, R0
11| STR R1, .Pixel367
12| MOV R2, #1
13| BL delay
14| POP {LR,R1,R2}
15| RET
16|delay:
17| PUSH {R3,R4,R5,R6,R7}
18| MOV R3, R2
19| LDR R4, .Time
20|timer:
21| LDR R5, .Time
22| SUB R6, R5, R4
23| CMP R6, R3
24| BLT timer
25| POP {R3,R4,R5,R6,R7}
26| RET
```

Processor state:

Register	Value
PC	0x00000000
LR	0x00000000
SP	0x00100000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0x00000000
R5	0x00000000
R4	0x00000000
R3	0x00000000
R2	0x00000000
R1	0x00000000
R0	0x00000000

Status bits: NZCV 0000

Count: 0

Current: 0

Instruction: 0

Input/Output: Stop done, edit & Submit, RUN/STEP or alter memory

Q10.1 (c):

The above code uses indirect addressing since it may store the exact value of R2 in [R4], containing a memory location and its value. In the code, 4 bytes are added to the pixel each time until it reaches 80, and data is moved to the memory location (4 bytes, 32 bits). R1 is the base address; if we add a value in R3, it will be added to R1, which holds the address and may be used to create a new pointer to the subsequent pixel.

Q10.2:

### Program

```

1|flash:
2|    MOV R0, #.green
3|    BL drawpixel
4|    MOV R0, #.white
5|    BL drawpixel
6|    MOV R0, #.green
7|    BL drawpixel
8|    MOV R0, #.white
9|    BL drawpixel
10|   MOV R0, #.green
11|   BL drawpixel
12|   MOV R0, #.white
13|   BL drawpixel
14|   MOV R2, #2
15|   BL delay
16|   B flash
17|   HALT
18|drawpixel:
19|   PUSH {LR,R1,R2}  //R3 is basically local variable
20|   MOV R1, R0
21|   STR R1, .Pixel367
22|   MOV R2, #1
23|   BL delay
24|   POP {LR,R1,R2}
25|   RET
26|delay:
27|   PUSH {R1,R4,R5,R6}
28|   MOV R1, R2
29|   LDR R4, .Time
30|timer:
31|   LDR R5, .Time
32|   SUB R6, R5, R4
33|   CMP R6, R1
34|   BLT timer
35|   POP {R1,R4,R5,R6}
36|   RET

```

### Processor

PC	0x00000000
LR	0x00000000
SP	0x00010000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0x00000000
R5	0x00000000
R4	0x00000000
R3	0x00000000
R2	0x00000000
R1	0x00000000
R0	0x00000000

Count:

Current Instruction:

Status bits: **NZCV**  
**0000**

### Input/Output

Stop done, edit & Submit, RUN/STEP or alter memory

Q10.3:

### Program

```

1|flash:
2|    MOV R5, #0
3|    BL flashpattern
4|    CMP R5, R3
5|    BLT rapidflash
6|    BEQ cont
7|cont:
8|    MOV R7, R2
9|    BL delay
10|   HALT
11|drawpixel:
12|   PUSH {LR,R1,R2}  //R3 is basically local variable
13|   MOV R1, R0
14|   STR R1, .Pixel367
15|   MOV R2, R7
16|   BL delay
17|   POP {LR,R1,R2}
18|   RET
19|delay:
20|   PUSH {R1,R4,R5,R6}
21|   MOV R1, R2
22|   LDR R4, .Time
23|timer:
24|   LDR R5, .Time
25|   SUB R6, R5, R4
26|   CMP R6, R1
27|   BLT timer
28|   POP {R1,R4,R5,R6}
29|   RET
30|flashpattern:
31|   PUSH {R1,R2}
32|   MOV R1, #flashnum
33|   STR R1, .WriteString
34|   LDR R1, .InputNum
35|   STR R1, .WriteUnsignednum
36|   MOV R2, #delaynum

```

### Processor

PC	0
LR	1048576
SP	0
R12	0
R11	0
R10	0
R9	0
R8	0
R7	0
R6	0
R5	0
R4	0
R3	0
R2	0
R1	0
R0	0

Count:

Current Instruction:

Status bits: **NZCV**  
**0000**

### Input/Output

Saving File

### Memory

	0x0	0x4	0x8	0xc
0x0000	-476033024	-352321515	-514523133	-1174405088
0x0001	184549375	-509579262	-352321529	-520093584
0x0002	-382910458	-509603840	-451995788	-509599737
0x0003	-352321535	-390250490	-509546482	-382926734
0x0004	-509603838	-450936568	-450932468	-532324348
0x0005	-514457599	-1157627909	-390266766	-509546482
0x0006	-382926842	-476046335	-451997360	-450948756
0x0007	-451997340	-486530797	-451993248	-450944644
0x0008	-451993228	-509595647	-509591550	-390266874
0x0009	-509546482	-382910463	-476051198	-335544353
0x000a	218103807	-335544355	-494579711	-390250495
0x000b	-509546482	0	0	0
0x000c	0	0	0	0
0x000d	0	0	0	0
0x000e	0	0	0	0
0x000f	0	0	0	0
0x0010	544698184	2037277037	1634493984	1936025715
0x0011	167780410	544698184	2037277037	1818584096
0x0012	980646241	32	0	0
0x0013	0	0	0	0
0x0014	0	0	0	0
0x0015	0	0	0	0
0x0016	0	0	0	0
0x0017	0	0	0	0
0x0018	0	0	0	0
0x0019	0	0	0	0
0x001a	0	0	0	0
0x001b	0	0	0	0
0x001c	0	0	0	0
0x001d	0	0	0	0
0x001e	0	0	0	0
0x001f	0	0	0	0

Decimal (signed)

Clear