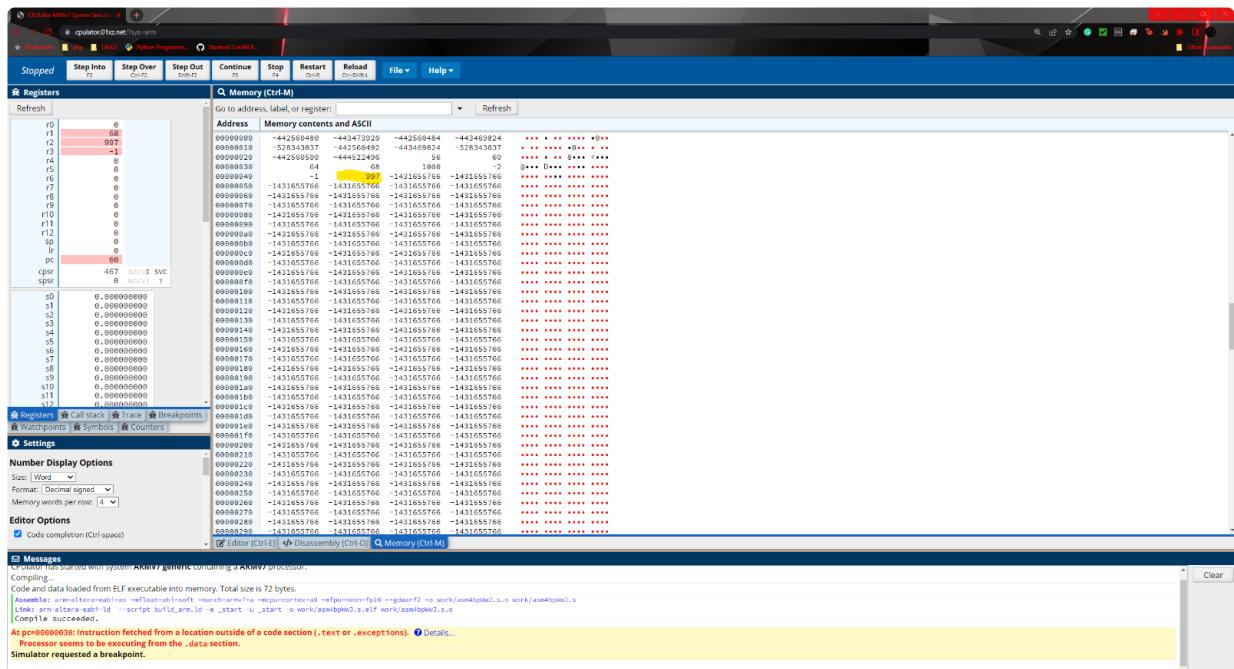
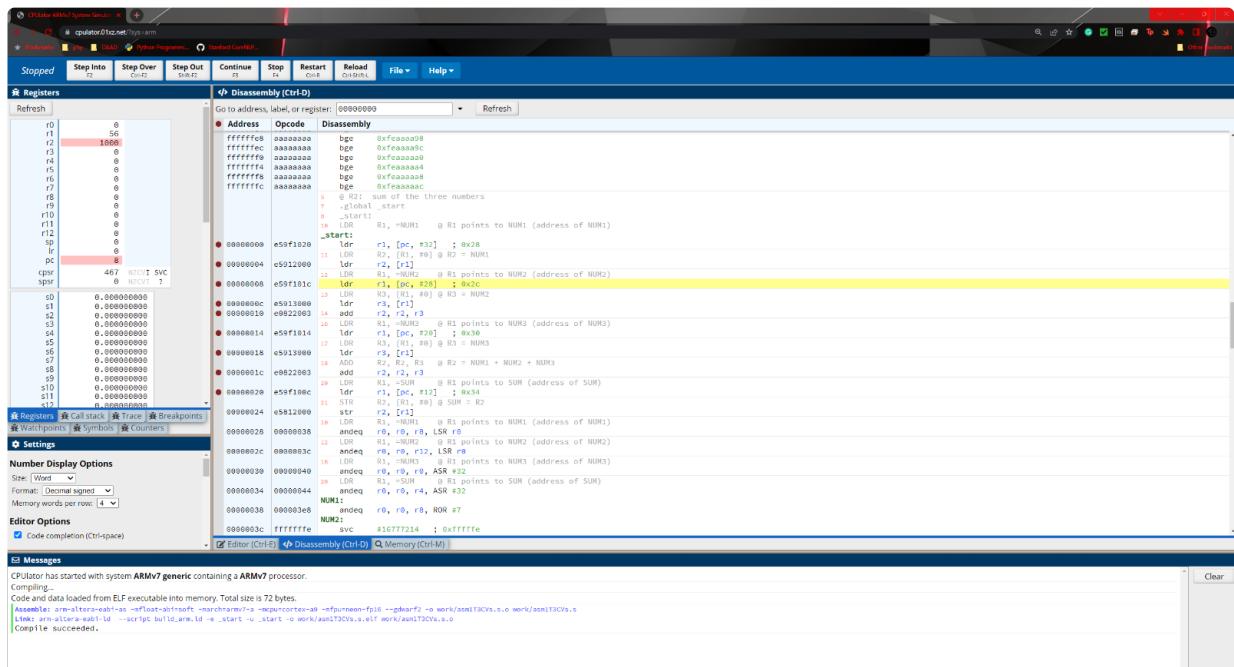


LAB 5 report ARM

5) 997



8)



The screenshot shows the CPULator interface with several panes. The top navigation bar includes tabs for CPU, Registers, Watchpoints, Symbols, Counters, Settings, Editor Options, and Messages. The CPU tab is active, displaying assembly code for a routine named 'sum'. The Registers pane shows the state of various registers (r0-r17, sp, lr, pc, cpsr) with values like r0=0, r1=0, r2=1000, and pc=467. The Watchpoints, Symbols, and Counters panes are empty. The Editor Options pane has checkboxes for 'Code completion (Ctrl+space)' and 'Editor (Ctrl+E)'. The bottom pane displays the message 'CPULator has started with system ARMv7 generic containing a ARMv7 processor. Compiling...'. The status bar at the bottom right shows 'Clear'.

The screenshot shows the GDB interface for an ARM7 processor. The top menu bar includes 'File', 'Edit', 'Breakpoints', 'Registers', 'Watchpoints', 'Symbols', 'Counters', 'Settings', 'Editor Options', and 'Messages'. The 'Registers' tab is selected, displaying the current register values:

Register	Value
r0	0
r1	44
r2	998
r3	-1
r4	0
r5	0
r6	0
r7	0
r8	0
r9	0
r10	0
r11	0
r12	0
sp	0
lr	28
pcr	ARMV7 SVC
spcr	0 NZCVC ?

The assembly code window shows the following sequence of instructions:

```
0xfffffe8: aaaaaaaaaa bge 0xfeeaaab0
0xfffffe9: aaaaaaaaaa bge 0xfeeaaabd
0xfffffa0: aaaaaaaaaa bge 0xfeeaaaz0
0xfffffa1: aaaaaaaaaa bge 0xfeeaa44
0xfffffa2: aaaaaaaaaa bge 0xfeeaaab8
0xfffffa3: aaaaaaaaaa bge 0xfeeaaab9
0xfffffa4: aaaaaaaaaa sum = r1 + r2 + r3
0xfffffa5: aaaaaaaaaa @ R2: sum of the three numbers
0xfffffa6: aaaaaaaaaa +global_start
0xfffffa7: aaaaaaaaaa ldr R1, =NUM1 @ R1 points to NUM1 (address of NUM1)
0xfffffa8: e59ff028 .start:
0xfffffa9: aaaaaaaaaa ldr r1, [R1, #32] ; 0x28
0xfffffaa: aaaaaaaaaa ldr R2, [R1, #8] @ R2 = NUM1
0xfffffae: aaaaaaaaaa ldr r2, [R1]
0xfffffb0: aaaaaaaaaa ldr r3, [R1, #16] @ R3 points to NUM2 (address of NUM2)
0xfffffb1: aaaaaaaaaa ldr r1, [R1, #20] ; 0x2c
0xfffffb2: aaaaaaaaaa ldr r3, [R1, #40] @ R3 = NUM2
0xfffffb3: aaaaaaaaaa ldr r3, [R1]
0xfffffb4: aaaaaaaaaa ldr r1, [R1, #20] ; 0x30
0xfffffb5: aaaaaaaaaa ldr r2, [R1, #8] @ R2 = NUM3
0xfffffb6: aaaaaaaaaa ldr r3, [R1]
0xfffffb7: aaaaaaaaaa add R2, R2, R3 @ R2 = NUM1 + NUM2 + NUM3
0xfffffb8: aaaaaaaaaa add R2, R2, R3
0xfffffb9: aaaaaaaaaa ldr r1, [R1, #20] @ R1 points to SUM (address of SUM)
0xfffffb0: aaaaaaaaaa ldr r1, [R1, #12] ; 0x34
0xfffffb1: aaaaaaaaaa str R2, [R1, #8] @ SUM = R2
0xfffffb2: aaaaaaaaaa ldr r1, [R1, #20]
0xfffffb3: aaaaaaaaaa ldr r1, [R1, #40] @ R1 points to NUM1 (address of NUM1)
0xfffffb4: aaaaaaaaaa andeq r8, r8, r8, LSR #8
0xfffffb5: aaaaaaaaaa andeq r8, r8, r8, LSR #8 @ R8 points to NUM2 (address of NUM2)
0xfffffb6: aaaaaaaaaa andeq r8, r8, r12, LSR #9
0xfffffb7: aaaaaaaaaa ldr R1, =NUM3 @ R1 points to NUM3 (address of NUM3)
0xfffffb8: aaaaaaaaaa andeq r8, r8, r8, ASR #2
0xfffffb9: aaaaaaaaaa andeq r8, r8, r8, ASR #2 @ R8 points to SUM (address of SUM)
0xfffffb0: aaaaaaaaaa andeq r8, r8, r8, ROR #7
0xfffffb1: aaaaaaaaaa NUM2:
0xfffffb2: aaaaaaaaaa NUM2:
0xfffffb3: ffffffc0 svc +16777214 ; 0xffffffff
```

The bottom status bar indicates: 'CPU/Unit has started with system ARMv7 generic containing a ARMv7 processor. Compiling... Code and data loaded from ELF executable into memory. Total size is 72 bytes. Linker: arm-elf-gcc-4.4.2 --script build.elf.s -T start -n start -e work/aodT3Cv5.s.elf work/aodT3Cv5.s.o Compile succeeded.'

Registers

r0	0
r1	10
r2	-100
r3	1000
r4	0
r5	0
r6	0
r7	0
r8	0
r9	0
r10	0
r11	0
r12	0
sp	32
pc	407 NUM1 SVC
psr	0

Watchpoints

Symbols

Counters

Settings

Number Display Options

Editor Options

Code completion (Ctrl+Space)

Messages

CPUJutor has started with system ARMv7 generic containing a ARMv7 processor.

Compiling...

Code and data loaded from ELF executable into memory. Total size is 72 bytes.

Assemble: arm-elf-as -mfloatabi=soft -march=armv7-a -mcps=cortex-a8 -mfpu=neon-float --o=hellof2.o -w /work/armST3209.s.o -w /work/armST3209.s.d

Link: arm-elf-gcc-4.8 -mfloatabi=soft -script build_arm_ld -e _start -u _start -u /work/armST3209.o -e /work/armST3209.o.o

Compile succeeded.

9) new values:

num1: .word 10

num2: .word -100

num3: .word 1000

Before:

Registers

r0	10
r1	-100
r2	1000
r3	0
r4	0
r5	0
r6	0
r7	0
r8	0
r9	0
r10	0
r11	0
r12	0
sp	0
pc	0
psr	0

Watchpoints

Symbols

Counters

Settings

Number Display Options

Editor Options

Code completion (Ctrl+Space)

Messages

CPUJutor has started with system ARMv7 generic containing a ARMv7 processor.

Compiling...

Code and data loaded from ELF executable into memory. Total size is 72 bytes.

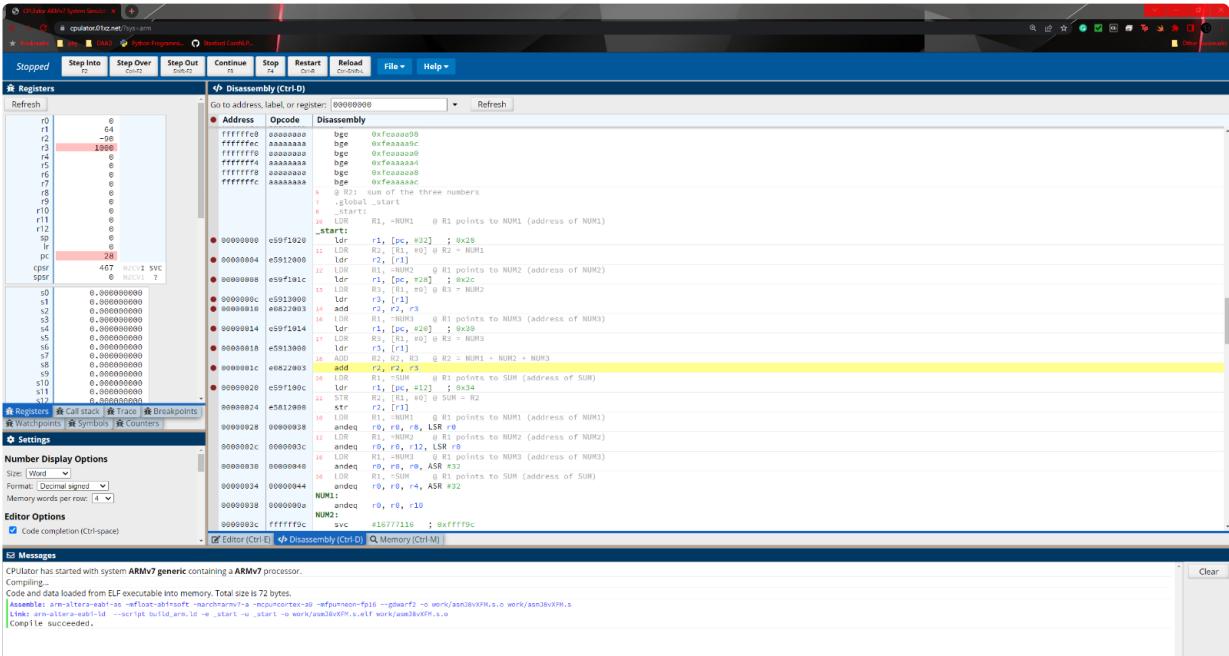
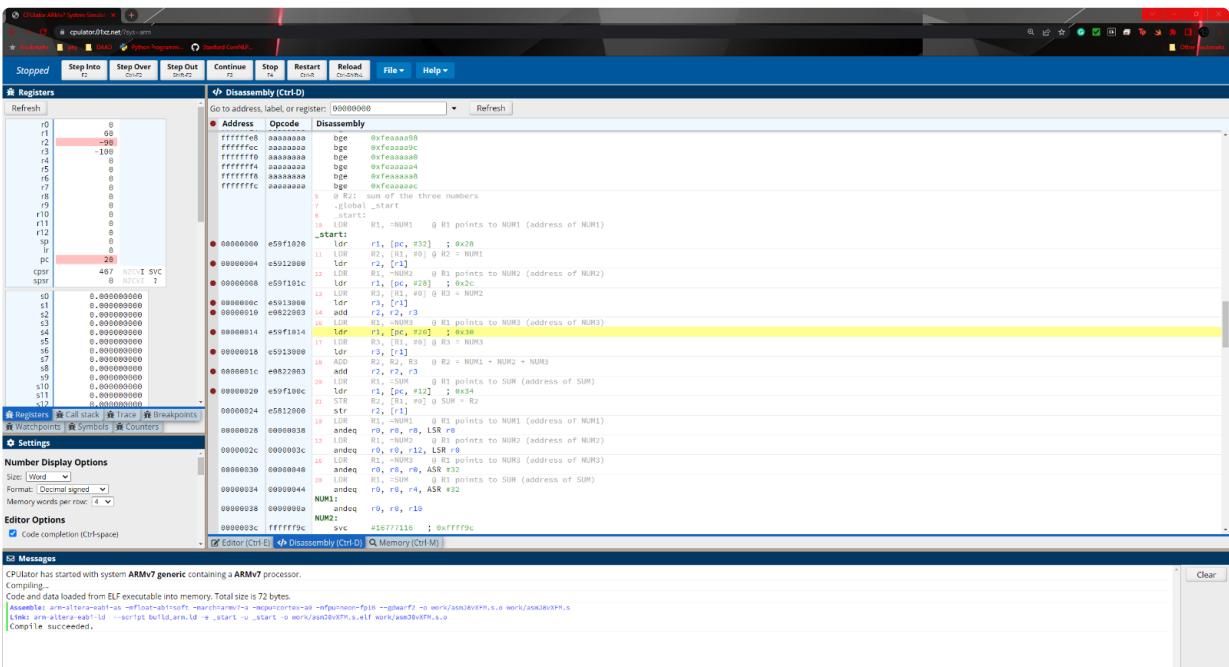
Assemble: arm-elf-as -mfloatabi=soft -march=armv7-a -mcps=cortex-a8 -mfpu=neon-float --o=hellof2.o -w /work/armST3209.s.o -w /work/armST3209.s.d

Link: arm-elf-gcc-4.8 -mfloatabi=soft -script build_arm_ld -e _start -u _start -u /work/armST3209.o -e /work/armST3209.o.o

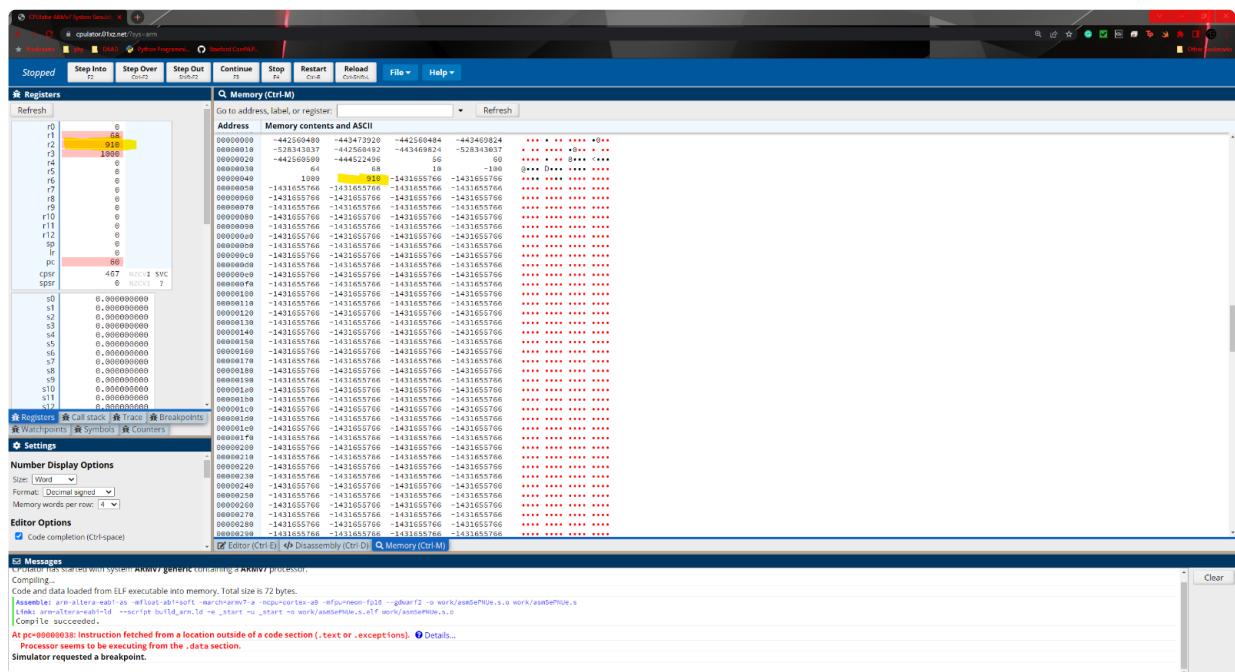
Compile succeeded.

During:

The screenshot shows the QEMU ARMv7 generic monitor interface. The top menu bar includes 'File', 'Edit', 'Registers', 'Call stack', 'Trace', 'Breakpoints', 'Watchpoints', 'Symbols', 'Counters', 'Settings', 'Editor Options', and 'Messages'. The main window displays the assembly and registers. The assembly pane shows the disassembly of memory starting at address 0x00000000, with labels like _start, _LDR, and _add. The registers pane shows various CPU registers (r0-r15, sp, pc) and system registers (cpsr, spsr). The bottom status bar indicates the current editor tab is 'Disassembly (Ctrl-D)'.

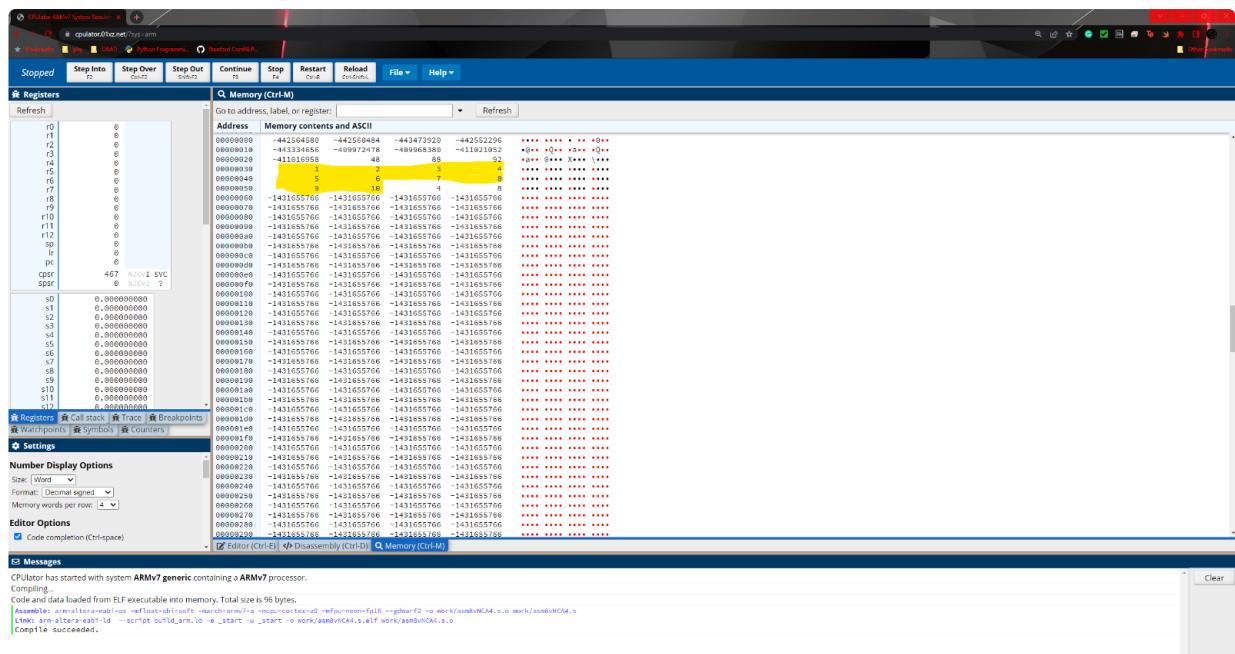


After:



10)

Before:



After:

