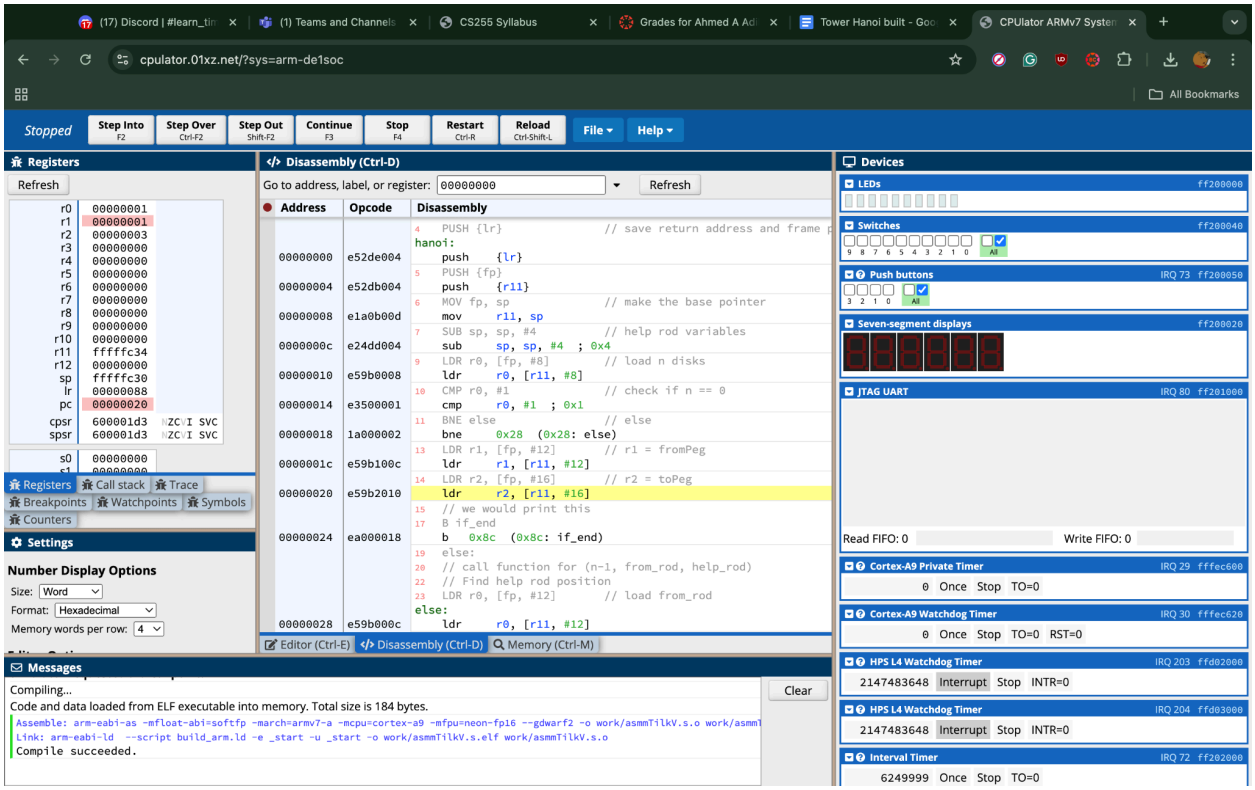
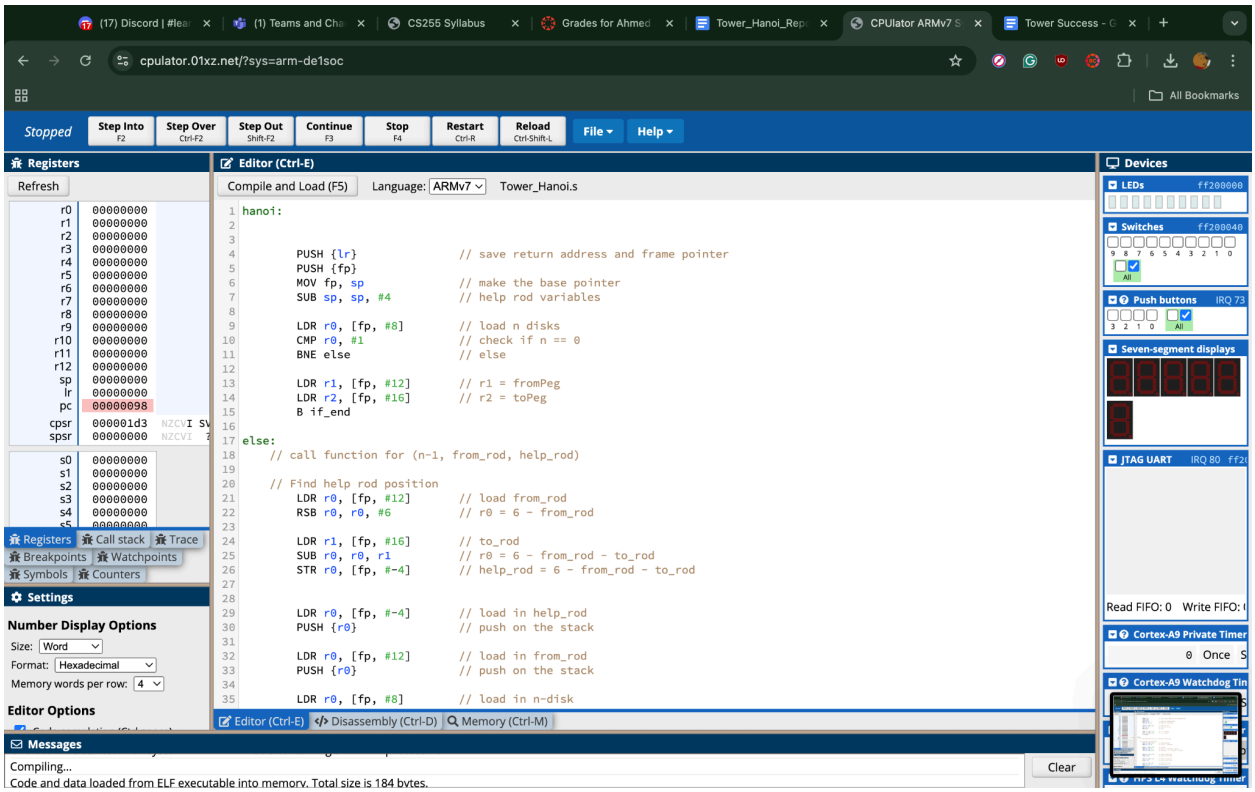
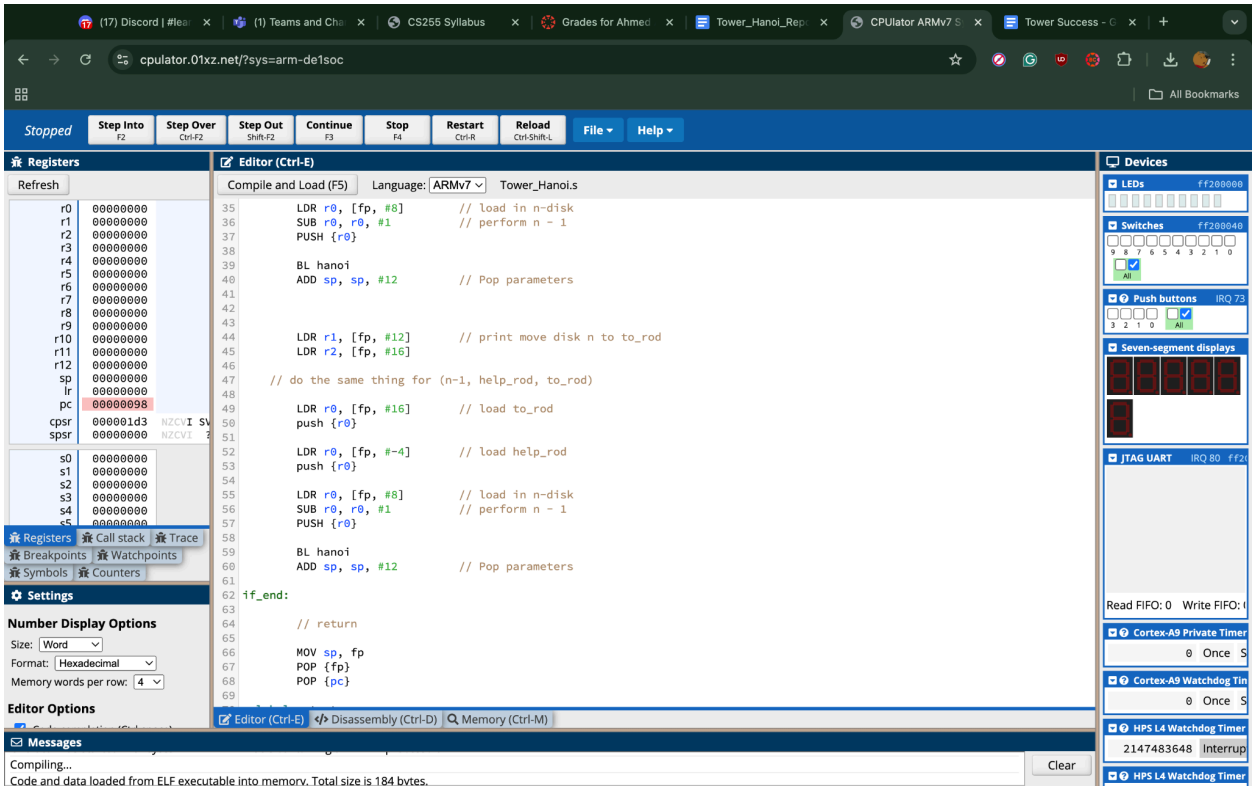
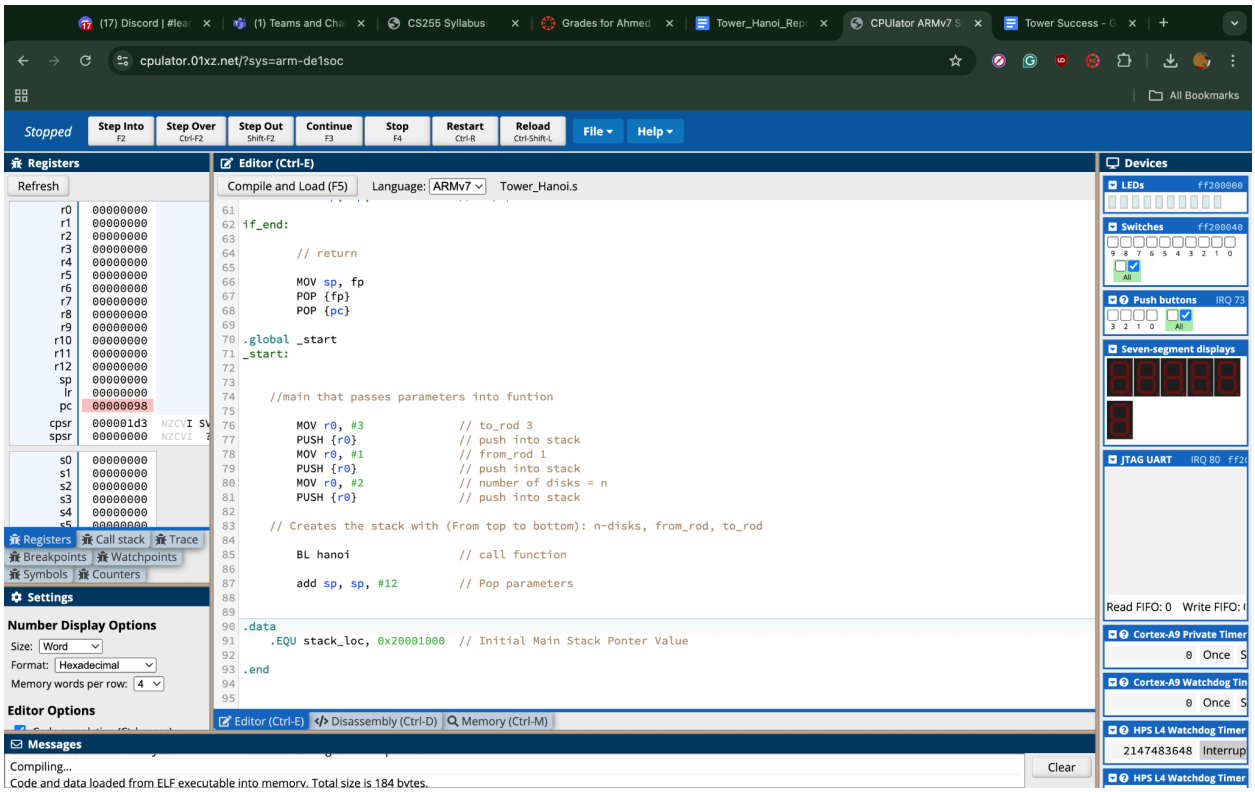

cpulator.01xz.net/?sys=arm-de1soc
Stopped
Step Into F2
Step Over Ctrl-F2
Step Out Shift-F2
Continue F3
Stop F4
Restart Ctrl-R
Reload Ctrl-Shift-L
File
Help
Registers
Refresh
r0 00000001
r1 00000001
r2 00000003
r3 00000000
r4 00000000
r5 00000000
r6 00000000
r7 00000000
r8 00000000
r9 00000000
r10 00000000
r11 fffffc34
r12 00000000
sp fffffc30
lr 00000088
pc 00000020
cpsr 600001d3 NZCVI SVC
spsr 600001d3 NZCVI SVC
s0 00000000
Registers
Call stack
Trace
Breakpoints
Watchpoints
Symbols
Counters
Settings
Number Display Options
Size: Word
Format: Hexadecimal
Memory words per row: 4
Disassembly (Ctrl-D)
Go to address, label, or register: 00000000
Refresh
Address
Opcode
Disassembly
4 PUSH {lr} // save return address and frame p
hanoi:
00000000 e52de004 push {lr}
5 PUSH {fp}
00000004 e52db004 push {r11}
6 MOV fp, sp // make the base pointer
00000008 e1a0b00d mov r11, sp
7 SUB sp, sp, #4 // help rod variables
0000000c e24dd004 sub sp, sp, #4 ; 0x4
9 LDR r0, [fp, #8] // load n disks
00000010 e59b0008 ldr r0, [r11, #8]
10 CMP r0, #1 // check if n == 0
00000014 e3500001 cmp r0, #1 ; 0x1
11 BNE else // else
00000018 1a000002 bne 0x28 (0x28: else)
13 LDR r1, [fp, #12] // r1 = fromPeg
0000001c e59b100c ldr r1, [r11, #12]
14 LDR r2, [fp, #16] // r2 = toPeg
00000020 e59b2010 ldr r2, [r11, #16]
15 // we would print this
17 B if_end
00000024 ea000018 b 0x8c (0x8c: if_end)
19 else:
20 // call function for (n-1, from_rod, help_rod)
22 // Find help rod position
23 LDR r0, [fp, #12] // load from_rod
else:
00000028 e59b000c ldr r0, [r11, #12]
Editor (Ctrl-E)
Disassembly (Ctrl-D)
Memory (Ctrl-M)
Devices
LEDs ff200000
Switches ff200040
Push buttons IRQ 73 ff200050
Seven-segment displays ff200020
JTAG UART IRQ 80 ff201000
Read FIFO: 0
Write FIFO: 0
Cortex-A9 Private Timer IRQ 29 fffec600
0 Once Stop TO=0
Cortex-A9 Watchdog Timer IRQ 30 fffec620
0 Once Stop TO=0 RST=0
HPS L4 Watchdog Timer IRQ 203 ffd02000
2147483648 Interrupt Stop INTR=0
HPS L4 Watchdog Timer IRQ 204 ffd03000
2147483648 Interrupt Stop INTR=0
Interval Timer IRQ 72 ff202000
6249999 Once Stop TO=0
Messages
Clear
Compiling...
Code and data loaded from ELF executable into memory. Total size is 184 bytes.
Assemble: arm-eabi-as -mfloat-abi=softfp -march=armv7-a -mcpu=cortex-a9 -mfpu=neon-fp16 --gdwarf2 -o work/asmmTilkV.s.o work/asmmT
Link: arm-eabi-ld --script build_arm.ld -e _start -u _start -o work/asmmTilkV.s.elf work/asmmTilkV.s.o
Compile succeeded.


cpulator.01xz.net/?sys=arm-de1soc
Stopped
Step Into F2
Step Over Ctrl-F2
Step Out Shift-F2
Continue F3
Stop F4
Restart Ctrl-R
Reload Ctrl-Shift-L
File
Help
Registers
Refresh
r0 00000000
r1 00000000
r2 00000000
r3 00000000
r4 00000000
r5 00000000
r6 00000000
r7 00000000
r8 00000000
r9 00000000
r10 00000000
r11 00000000
r12 00000000
sp 00000000
lr 00000000
pc 00000098
cpsr 000001d3 NZCVI SV
spsr 00000000 NZCVI
s0 00000000
s1 00000000
s2 00000000
s3 00000000
s4 00000000
Registers
Call stack
Trace
Breakpoints
Watchpoints
Symbols
Counters
Settings
Number Display Options
Size: Word
Format: Hexadecimal
Memory words per row: 4
Editor Options
Editor (Ctrl-E)
Compile and Load (F5)
Language: ARMv7
Tower_Hanoi.s
1 hanoi:
4 PUSH {lr} // save return address and frame pointer
5 PUSH {fp}
6 MOV fp, sp // make the base pointer
7 SUB sp, sp, #4 // help rod variables
9 LDR r0, [fp, #8] // load n disks
10 CMP r0, #1 // check if n == 0
11 BNE else // else
13 LDR r1, [fp, #12] // r1 = fromPeg
14 LDR r2, [fp, #16] // r2 = toPeg
15 B if_end
17 else:
18 // call function for (n-1, from_rod, help_rod)
20 // Find help rod position
21 LDR r0, [fp, #12] // load from_rod
22 RSB r0, r0, #6 // r0 = 6 - from_rod
24 LDR r1, [fp, #16] // to_rod
25 SUB r0, r0, r1 // r0 = 6 - from_rod - to_rod
26 STR r0, [fp, #-4] // help_rod = 6 - from_rod - to_rod
29 LDR r0, [fp, #-4] // load in help_rod
30 PUSH {r0} // push on the stack
32 LDR r0, [fp, #12] // load in from_rod
33 PUSH {r0} // push on the stack
35 LDR r0, [fp, #8] // load in n-disk
Editor (Ctrl-E)
Disassembly (Ctrl-D)
Memory (Ctrl-M)
Devices
LEDs ff200000
Switches ff200040
Push buttons IRQ 73
Seven-segment displays
JTAG UART
Read FIFO: 0
Write FIFO:
Cortex-A9 Private Timer
0 Once
Cortex-A9 Watchdog Tin
Messages
Clear
Compiling...
Code and data loaded from ELF executable into memory. Total size is 184 bytes.


(17) Discord | #lear
(1) Teams and Cha
CS255 Syllabus
Grades for Ahmed
Tower_Hanoi_Repo
CPUlator ARMv7 S
Tower Success - G
cpulator.01xz.net/?sys=arm-de1soc
All Bookmarks
Stopped
Step Into F2
Step Over Ctrl-F2
Step Out Shift-F2
Continue F3
Stop F4
Restart Ctrl-R
Reload Ctrl-Shift-L
File
Help
Registers
Refresh
r0 00000000
r1 00000000
r2 00000000
r3 00000000
r4 00000000
r5 00000000
r6 00000000
r7 00000000
r8 00000000
r9 00000000
r10 00000000
r11 00000000
r12 00000000
sp 00000000
lr 00000000
pc 00000098
cpsr 000001d3 NZCVI SV
spsr 00000000 NZCVI
s0 00000000
s1 00000000
s2 00000000
s3 00000000
s4 00000000
Registers
Call stack
Trace
Breakpoints
Watchpoints
Symbols
Counters
Settings
Number Display Options
Size: Word
Format: Hexadecimal
Memory words per row: 4
Editor Options
Editor (Ctrl-E)
Compile and Load (F5)
Language: ARMv7
Tower_Hanoi.s
35 LDR r0, [fp, #8] // load in n-disk
36 SUB r0, r0, #1 // perform n - 1
37 PUSH {r0}
38
39 BL hanoi
40 ADD sp, sp, #12 // Pop parameters
41
42
43
44 LDR r1, [fp, #12] // print move disk n to to_rod
45 LDR r2, [fp, #16]
46
47 // do the same thing for (n-1, help_rod, to_rod)
48
49 LDR r0, [fp, #16] // load to_rod
50 push {r0}
51
52 LDR r0, [fp, #-4] // load help_rod
53 push {r0}
54
55 LDR r0, [fp, #8] // load in n-disk
56 SUB r0, r0, #1 // perform n - 1
57 PUSH {r0}
58
59 BL hanoi
60 ADD sp, sp, #12 // Pop parameters
61
62 if_end:
63
64 // return
65
66 MOV sp, fp
67 POP {fp}
68 POP {pc}
69
Editor (Ctrl-E)
Disassembly (Ctrl-D)
Memory (Ctrl-M)
Devices
LEDs ff200000
Switches ff200040
Push buttons IRQ 73
Seven-segment displays
JTAG UART IRQ 80
Read FIFO: 0 Write FIFO:
Cortex-A9 Private Timer
0 Once
Cortex-A9 Watchdog Tin
0 Once
HPS L4 Watchdog Timer
2147483648 Interrup
HPS L4 Watchdog Timer
Messages
Compiling...
Code and data loaded from ELF executable into memory. Total size is 184 bytes.
Clear


(17) Discord | #lear
(1) Teams and Cha
CS255 Syllabus
Grades for Ahmed
Tower_Hanoi_Repo
CPUlator ARMv7 S
Tower Success - G
cpulator.01xz.net/?sys=arm-de1soc
All Bookmarks
Stopped
Step Into F2
Step Over Ctrl-F2
Step Out Shift-F2
Continue F3
Stop F4
Restart Ctrl-R
Reload Ctrl-Shift-L
File
Help
Registers
Refresh
r0 00000000
r1 00000000
r2 00000000
r3 00000000
r4 00000000
r5 00000000
r6 00000000
r7 00000000
r8 00000000
r9 00000000
r10 00000000
r11 00000000
r12 00000000
sp 00000000
lr 00000000
pc 00000098
cpsr 000001d3 NZCVI SV
spsr 00000000 NZCVI
s0 00000000
s1 00000000
s2 00000000
s3 00000000
s4 00000000
Registers
Call stack
Trace
Breakpoints
Watchpoints
Symbols
Counters
Settings
Number Display Options
Size: Word
Format: Hexadecimal
Memory words per row: 4
Editor Options
Editor (Ctrl-E)
Compile and Load (F5)
Language: ARMv7
Tower_Hanoi.s
61
62 if_end:
63
64 // return
65
66 MOV sp, fp
67 POP {fp}
68 POP {pc}
69
70 .global _start
71 _start:
72
73
74 //main that passes parameters into funtion
75
76 MOV r0, #3 // to_rod 3
77 PUSH {r0} // push into stack
78 MOV r0, #1 // from_rod 1
79 PUSH {r0} // push into stack
80 MOV r0, #2 // number of disks = n
81 PUSH {r0} // push into stack
82
83 // Creates the stack with (From top to bottom): n-disks, from_rod, to_rod
84
85 BL hanoi // call function
86
87 add sp, sp, #12 // Pop parameters
88
89
90 .data
91 .EQU stack_loc, 0x20001000 // Initial Main Stack Ponter Value
92
93 .end
94
95
Editor (Ctrl-E)
Disassembly (Ctrl-D)
Memory (Ctrl-M)
Devices
LEDs ff200000
Switches ff200040
Push buttons IRQ 73
Seven-segment displays
JTAG UART IRQ 80
Read FIFO: 0 Write FIFO:
Cortex-A9 Private Timer
0 Once
Cortex-A9 Watchdog Tin
0 Once
HPS L4 Watchdog Timer
2147483648 Interrup
HPS L4 Watchdog Timer
Messages
Compiling...
Code and data loaded from ELF executable into memory. Total size is 184 bytes.
Clear