

Ethan Woycehoski

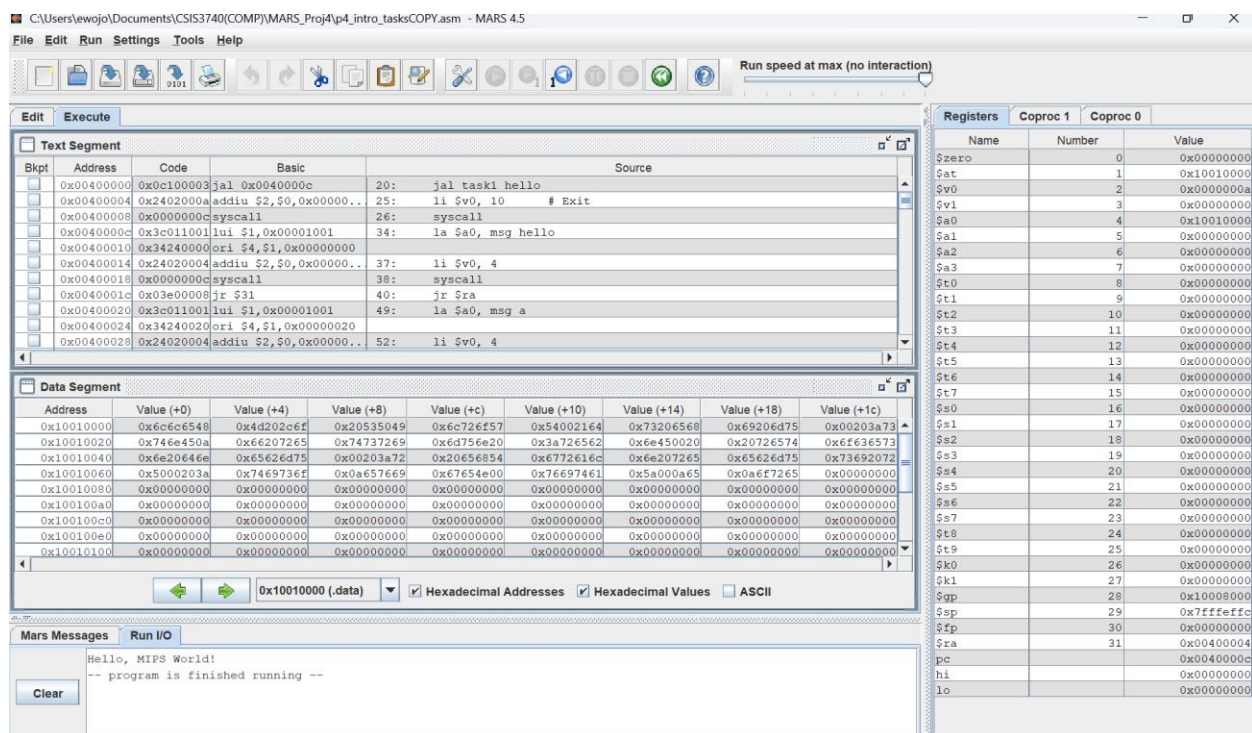
CSIS 3740

Prof. Jiang

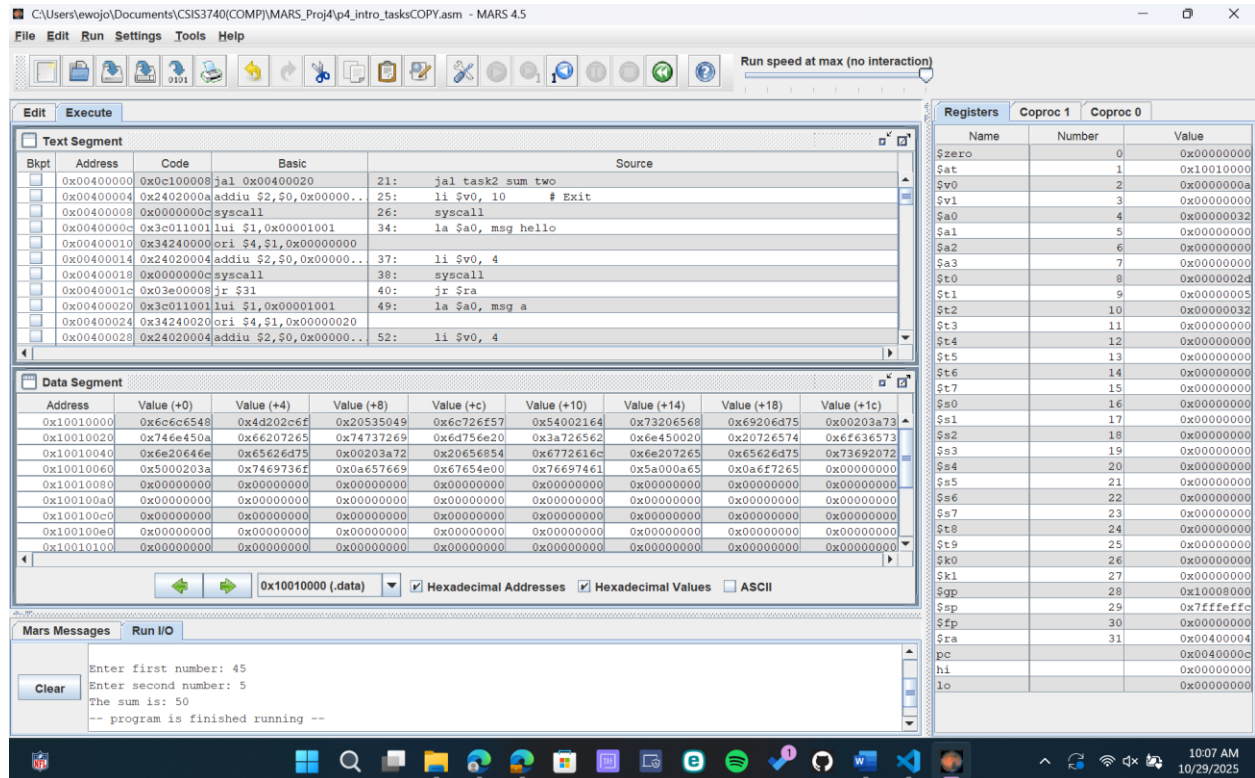
10/31/2025

Project Four Report

In this project, I learned about MARS and basic MIPS assembly. I completed all four of the tasks outlined in the instructions. I was able to print statements like, “Hello, MIPS World!” to the console.



I was able to read two integers from the user and display their sum back to the console.



I also successfully compared two numbers read from the console and determined whether a number was positive, negative, or zero when input by the user.

C:\Users\ewojp\Documents\CSIS3740(COMP)\MARS_Proj4\p4_intro_tasksCOPY.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x0c10001f	jal 0x0040007c	22: jal task3 compare two
	0x00400004	0x2402000a	addiu \$2,\$0,0x00000...	25: li \$v0, 10 # Exit
	0x00400008	0x0000000c	syscall	26: syscall
	0x0040000c	0x3c011001	lui \$1,0x00001001	34: la \$a0, msg hello
	0x00400010	0x34240000	ori \$4,\$1,0x00000000	
	0x00400014	0x2402000a	addiu \$2,\$0,0x00000...	37: li \$v0, 4
	0x00400018	0x0000000c	syscall	38: syscall
	0x0040001c	0x03e00008	jr \$31	40: jr \$ra
	0x00400020	0x3c011001	lui \$1,0x00001001	49: la \$a0, msg a
	0x00400024	0x34240020	ori \$4,\$1,0x00000020	
	0x00400028	0x2402000a	addiu \$2,\$0,0x00000...	52: li \$v0, 4

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x6c6c6548	0x4d202c6f	0x20535049	0x6c726f57	0x54002164	0x73206568	0x69206d75	0x00203a73
0x10010020	0x746e450a	0x66207265	0x74737269	0x6d756e20	0x3a726562	0x6e450020	0x20726574	0x6f636573
0x10010040	0x6e20646e	0x65626d75	0x00203a72	0x20656854	0x6772616c	0x6e207265	0x65626d75	0x73692072
0x10010060	0x5000203a	0x7469736f	0x0a657669	0x67654e00	0x76697461	0x5a00a65	0x0a6f7265	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

Enter first number: 34
Enter second number: 23
The larger number is: 34
-- program is finished running --

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x00000022
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000022
\$t4	12	0x00000017
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffffc0
\$fp	30	0x00000000
\$ra	31	0x00400004
pc		0x0040000c
hi		0x00000000
lo		0x00000000

C:\Users\ewojp\Documents\CSIS3740(COMP)\MARS_Proj4\p4_intro_tasksCOPY.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x0c10003f	jal 0x004000fc	23: jal task4 sign check
	0x00400004	0x2402000a	addiu \$2,\$0,0x00000...	25: li \$v0, 10 # Exit
	0x00400008	0x0000000c	syscall	26: syscall
	0x0040000c	0x3c011001	lui \$1,0x00001001	34: la \$a0, msg hello
	0x00400010	0x34240000	ori \$4,\$1,0x00000000	
	0x00400014	0x2402000a	addiu \$2,\$0,0x00000...	37: li \$v0, 4
	0x00400018	0x0000000c	syscall	38: syscall
	0x0040001c	0x03e00008	jr \$31	40: jr \$ra
	0x00400020	0x3c011001	lui \$1,0x00001001	49: la \$a0, msg a
	0x00400024	0x34240020	ori \$4,\$1,0x00000020	
	0x00400028	0x2402000a	addiu \$2,\$0,0x00000...	52: li \$v0, 4

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x6c6c6548	0x4d202c6f	0x20535049	0x6c726f57	0x54002164	0x73206568	0x69206d75	0x00203a73
0x10010020	0x746e450a	0x66207265	0x74737269	0x6d756e20	0x3a726562	0x6e450020	0x20726574	0x6f636573
0x10010040	0x6e20646e	0x65626d75	0x00203a72	0x20656854	0x6772616c	0x6e207265	0x65626d75	0x73692072
0x10010060	0x5000203a	0x7469736f	0x0a657669	0x67654e00	0x76697461	0x5a00a65	0x0a6f7265	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

Enter first number: 45
Positive
-- program is finished running --

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x10010063
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x0000002d
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffffc0
\$fp	30	0x00000000
\$ra	31	0x00400004
pc		0x0040000c
hi		0x00000000
lo		0x00000000

This project helped me to become familiar with the syntax and structure of a MIPS program, as

well as see how a computer executes instructions at a hardware level. In all, this project helped to deepen my understanding of how higher-level programming concepts, like input/output, conditional logic, and arithmetic, are translated and handled in lower-level instructions that directly control the processor.