LABORATORY REPORT

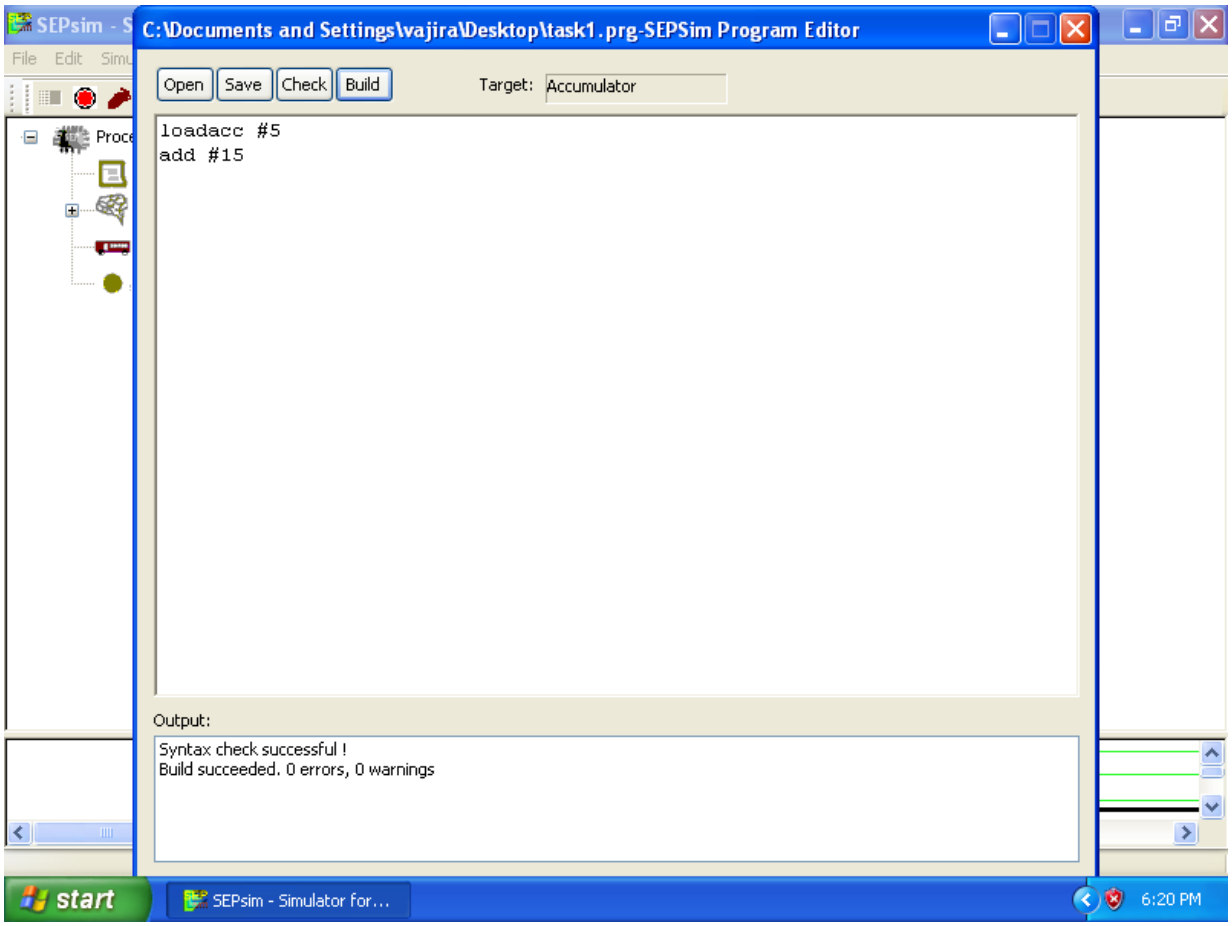
EEX – 3373 COMMUNICATION AND COMPUTER TECHNOLOGY

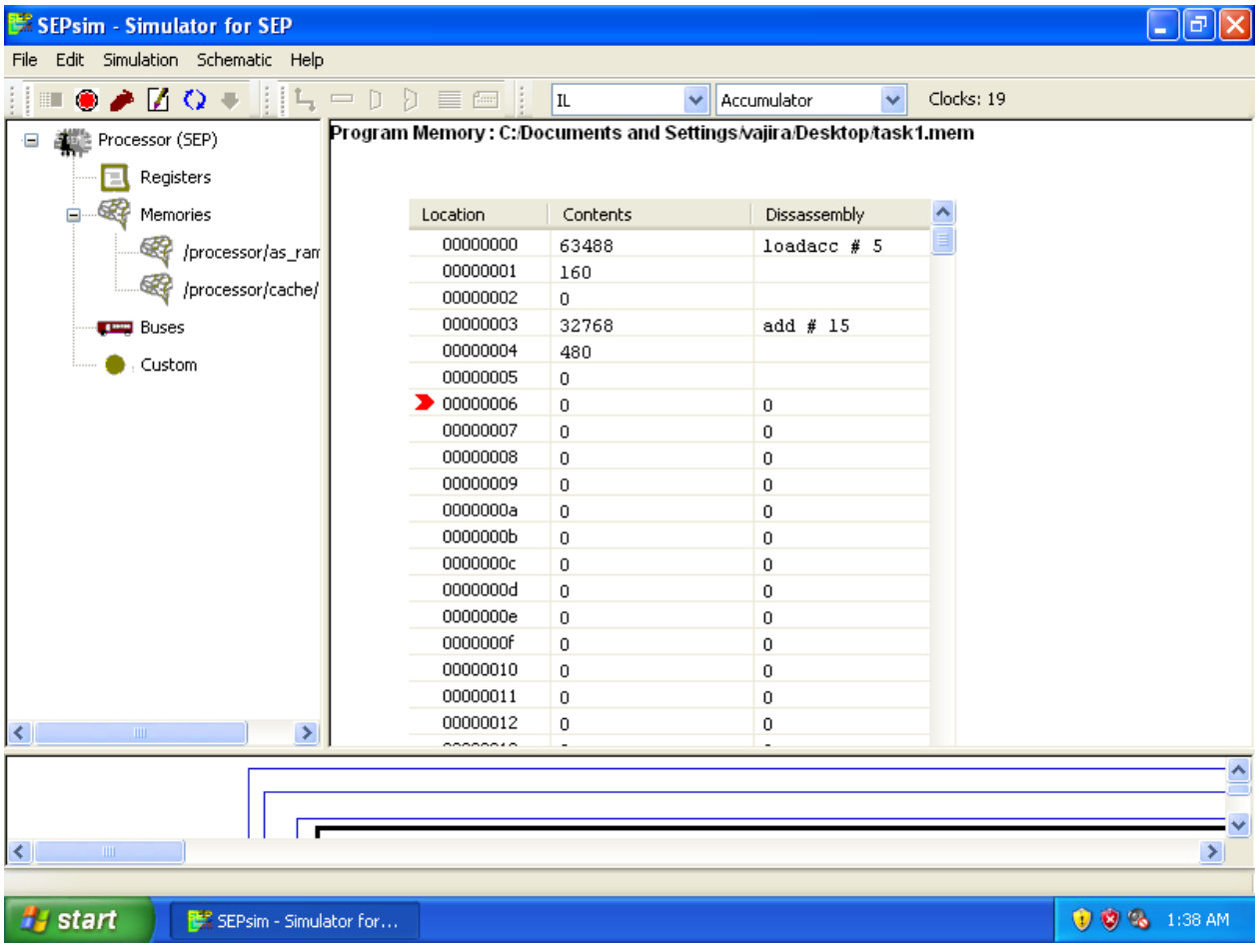
M.N.M. SAFRAN
S92064060

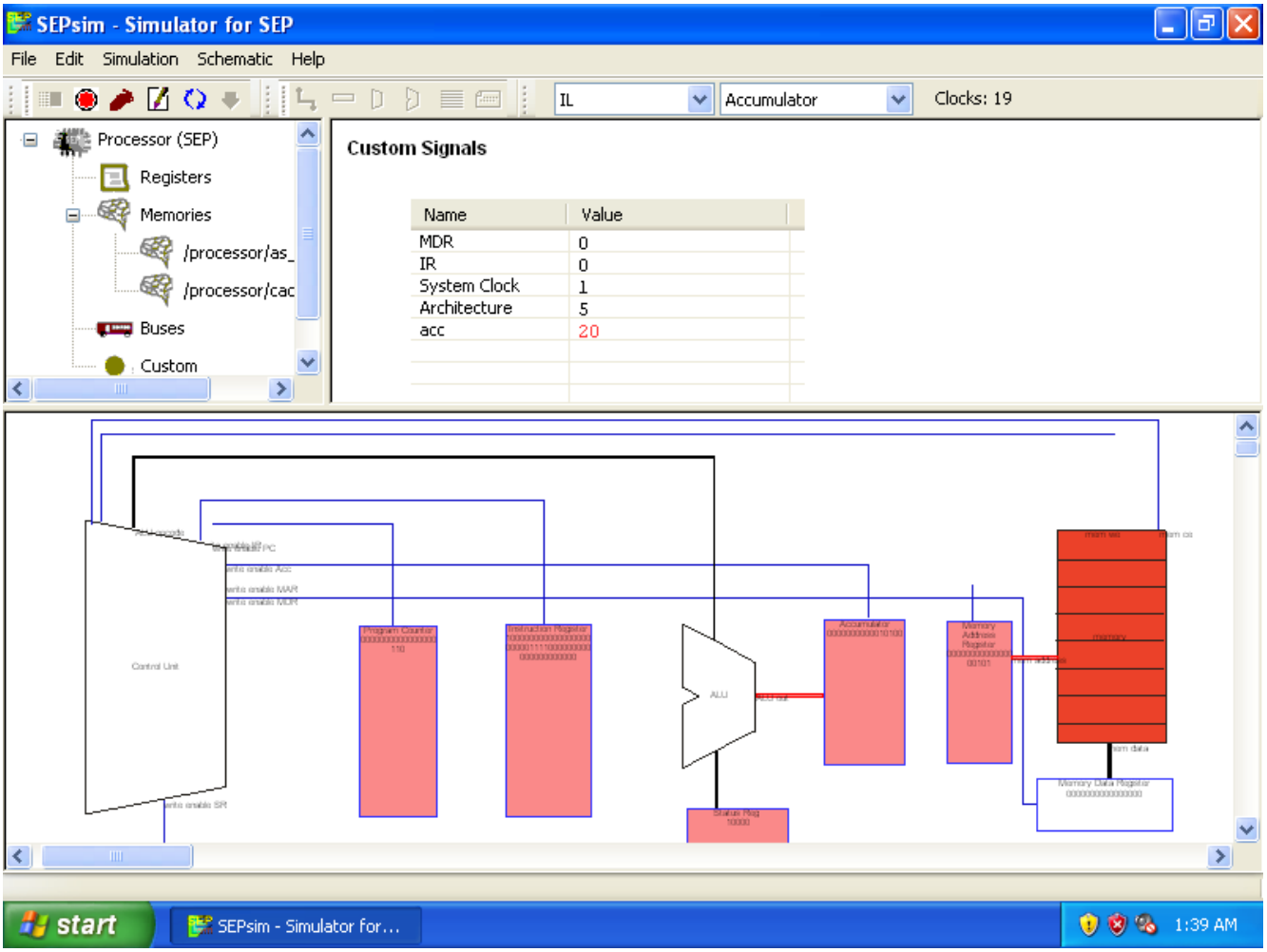
THE OPEN UNIVERSITY OF SRI LANKA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
BACHELOR OF SOFTWARE ENGINEERING

Academic Year 2021/2022
Lab – 02
Due Date: 05.08.2022

TASK 01


C:\Documents and Settings\vajira\Desktop\task1.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #5
add #15
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
start
SEPsim - Simulator for...
6:20 PM

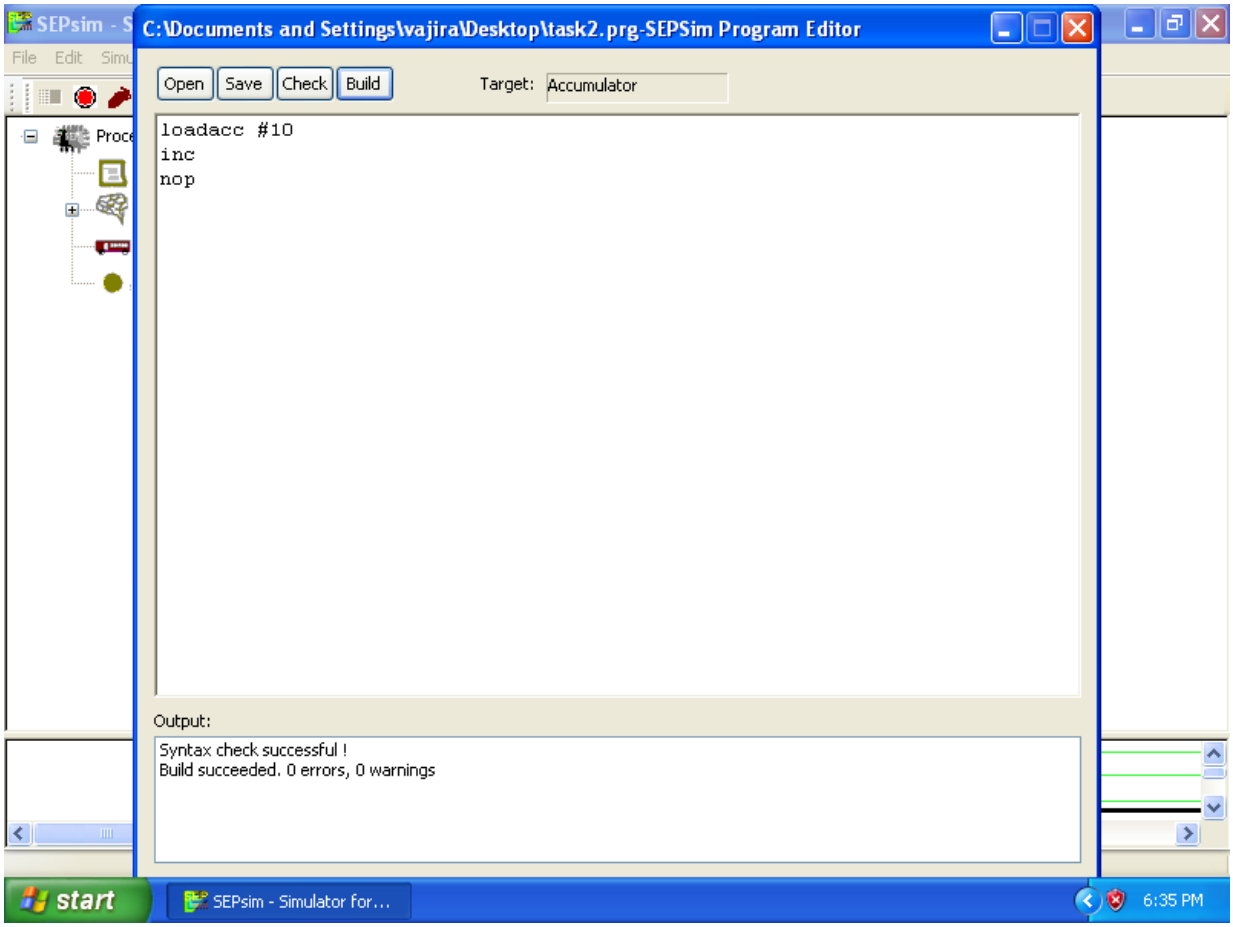

SEPsim - Simulator for SEP
File
Edit
Simulation
Schematic
Help
IL
Accumulator
Clocks: 19
Processor (SEP)
Registers
Memories
/processor/as_ram
/processor/cache/
Buses
Custom
Program Memory : C:/Documents and Settings/vajira/Desktop/task1.mem
Location
Contents
Disassembly
00000000 63488 loadacc # 5
00000001 160
00000002 0
00000003 32768 add # 15
00000004 480
00000005 0
00000006 0 0
00000007 0 0
00000008 0 0
00000009 0 0
0000000a 0 0
0000000b 0 0
0000000c 0 0
0000000d 0 0
0000000e 0 0
0000000f 0 0
00000010 0 0
00000011 0 0
00000012 0 0
start
SEPsim - Simulator for...
1:38 AM

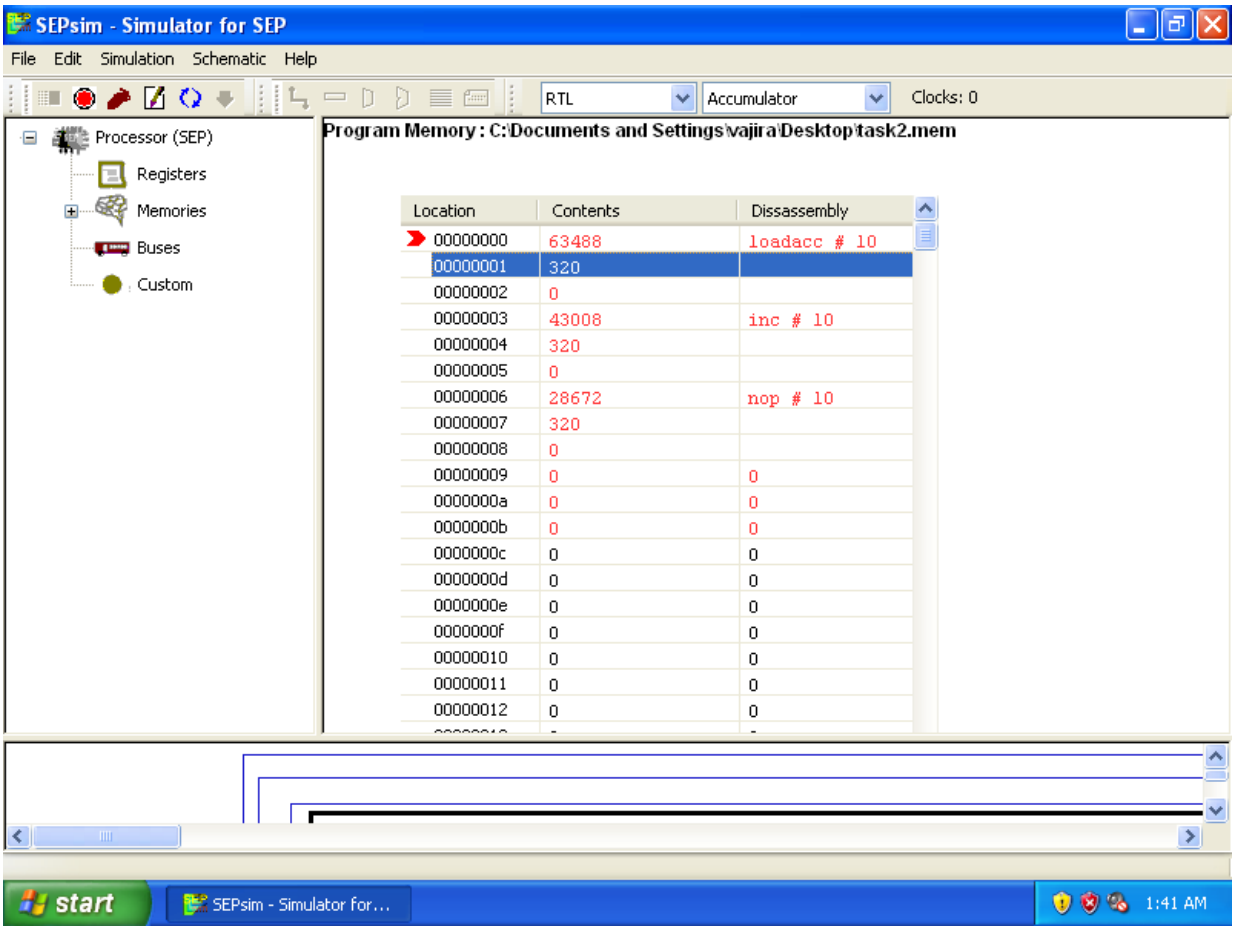


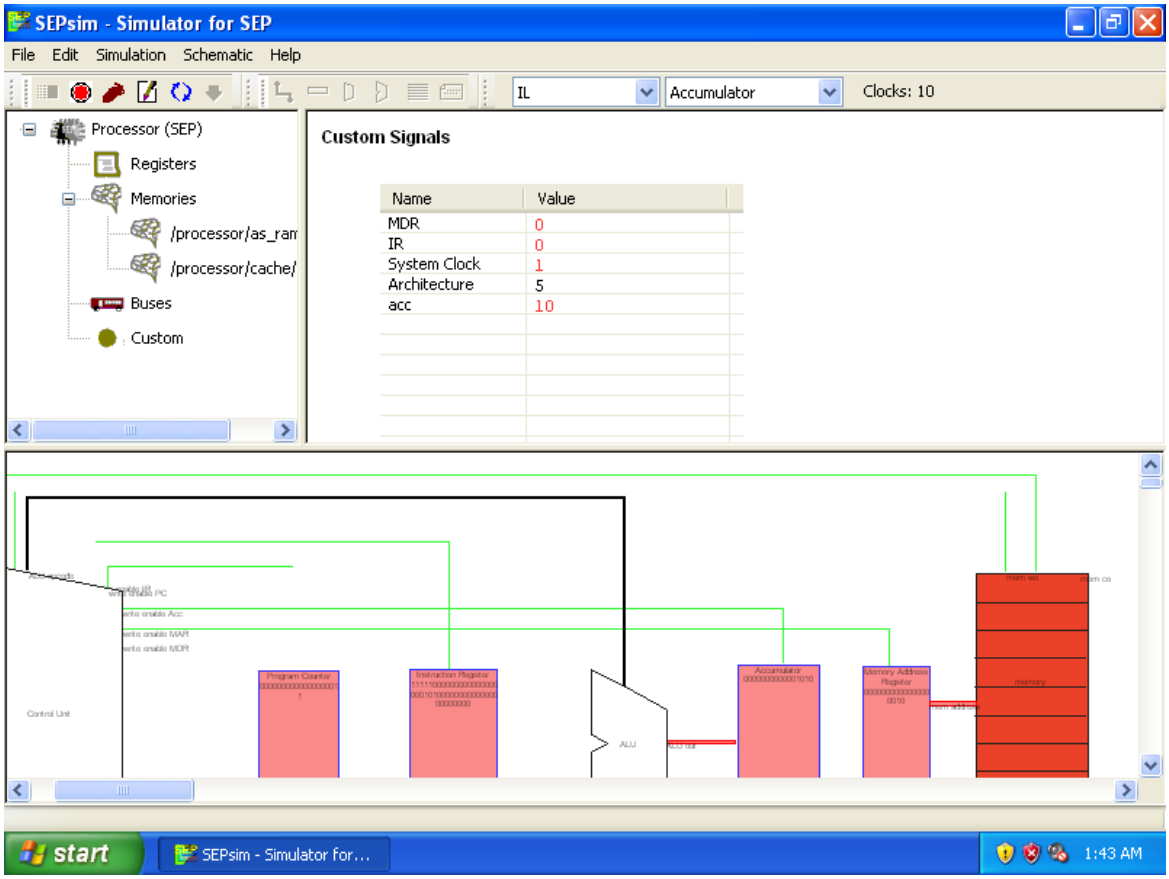
Discussion –

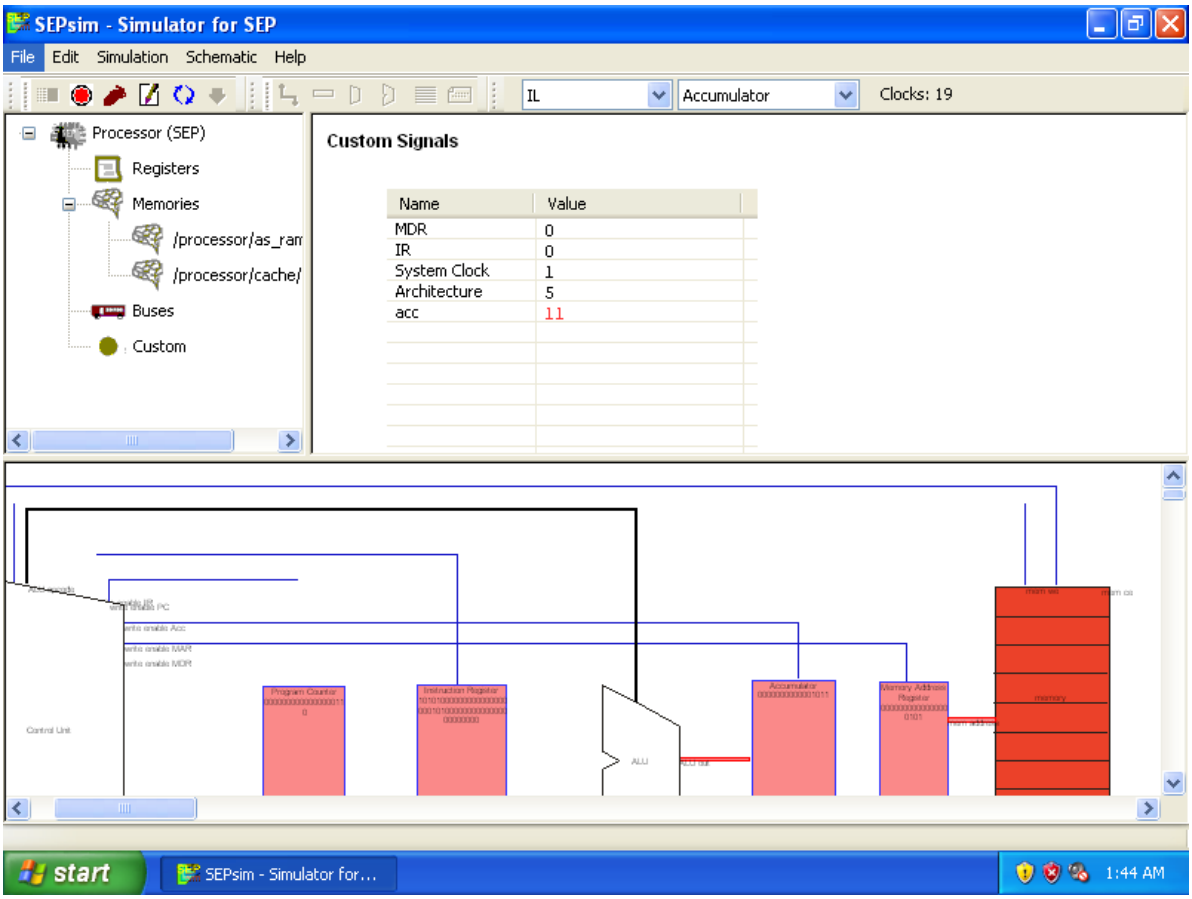
Here we load an integer number 5 and using ADD opcode to add a number.

TASK - 02


C:\Documents and Settings\vajira\Desktop\task2.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #10
inc
nop
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
start
SEPsim - Simulator for...
6:35 PM

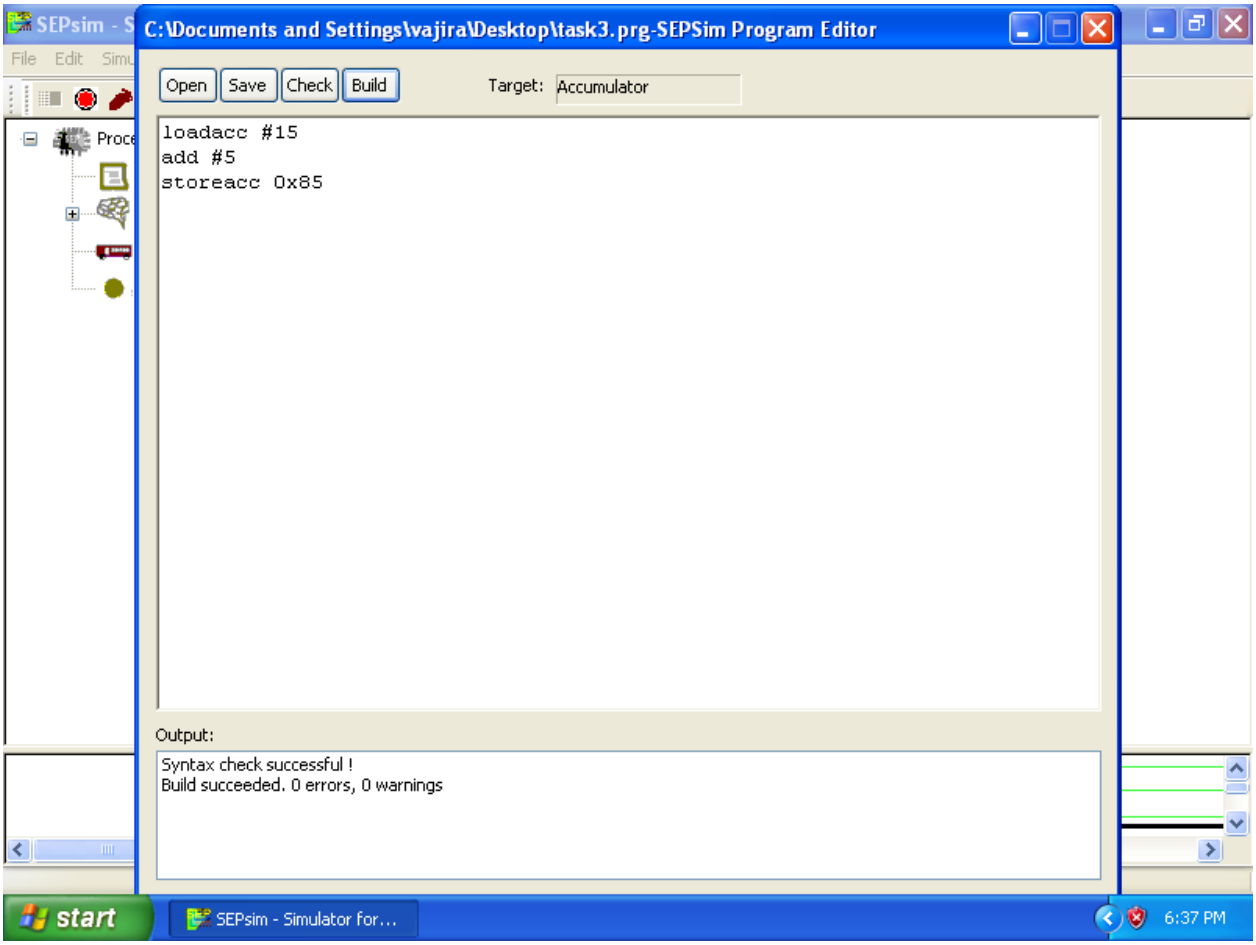

SEPsim - Simulator for SEP
File
Edit
Simulation
Schematic
Help
RTL
Accumulator
Clocks: 0
Processor (SEP)
Registers
Memories
Buses
Custom
Program Memory : C:\Documents and Settings\vajira\Desktop\task2.mem
Location
Contents
Dissassembly
00000000 63488 loadacc # 10
00000001 320
00000002 0
00000003 43008 inc # 10
00000004 320
00000005 0
00000006 28672 nop # 10
00000007 320
00000008 0
00000009 0 0
0000000a 0 0
0000000b 0 0
0000000c 0 0
0000000d 0 0
0000000e 0 0
0000000f 0 0
00000010 0 0
00000011 0 0
00000012 0 0
start
SEPsim - Simulator for...
1:41 AM

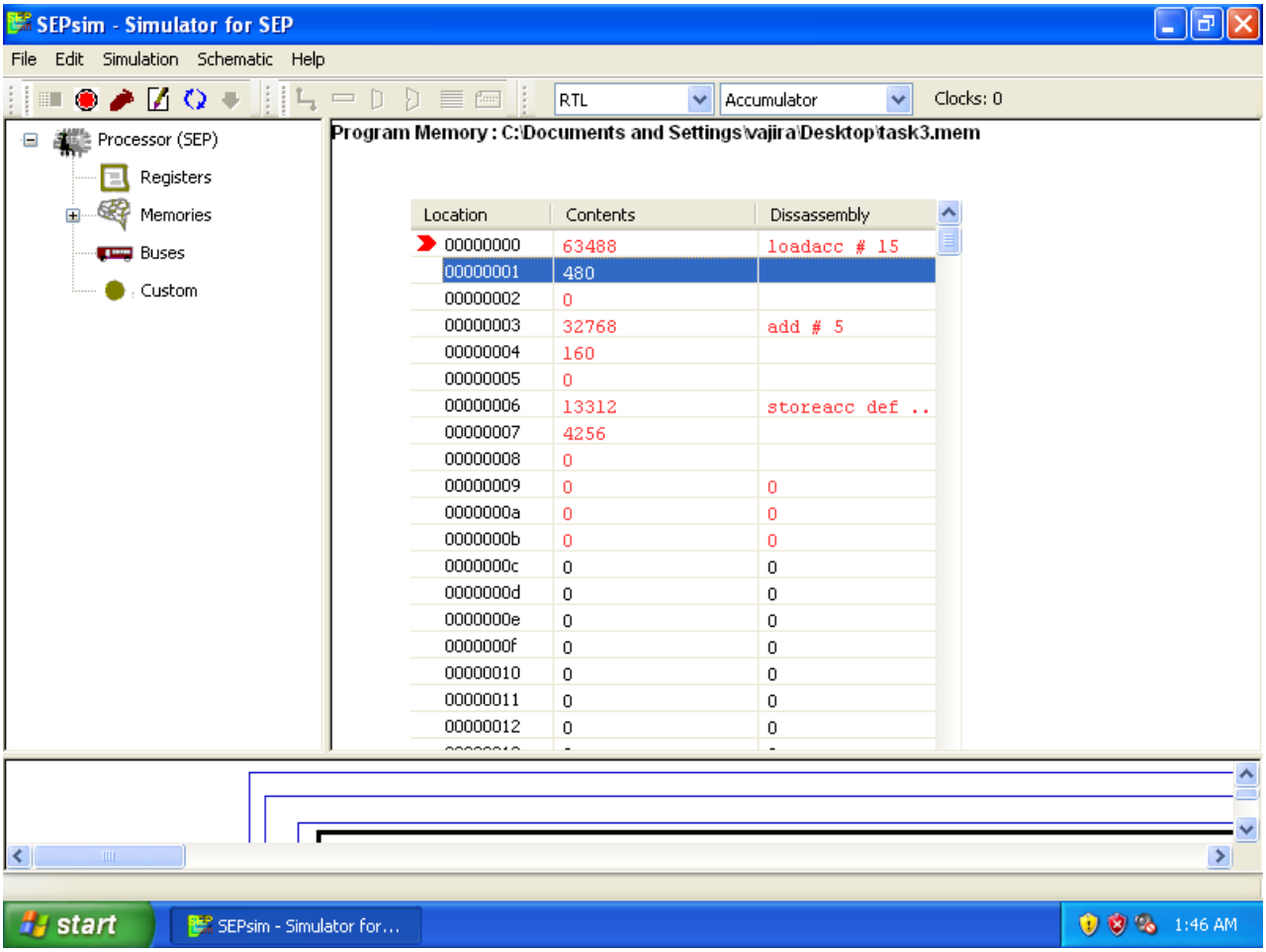


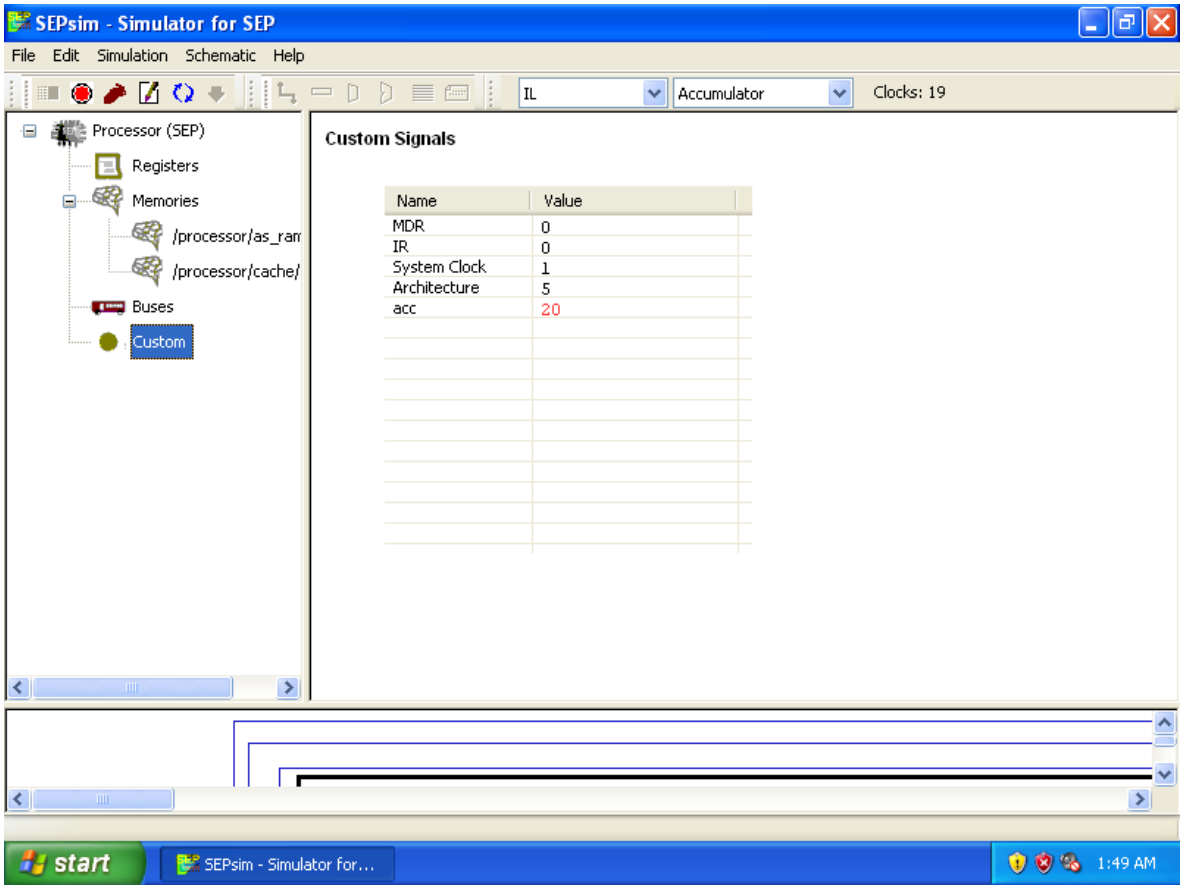


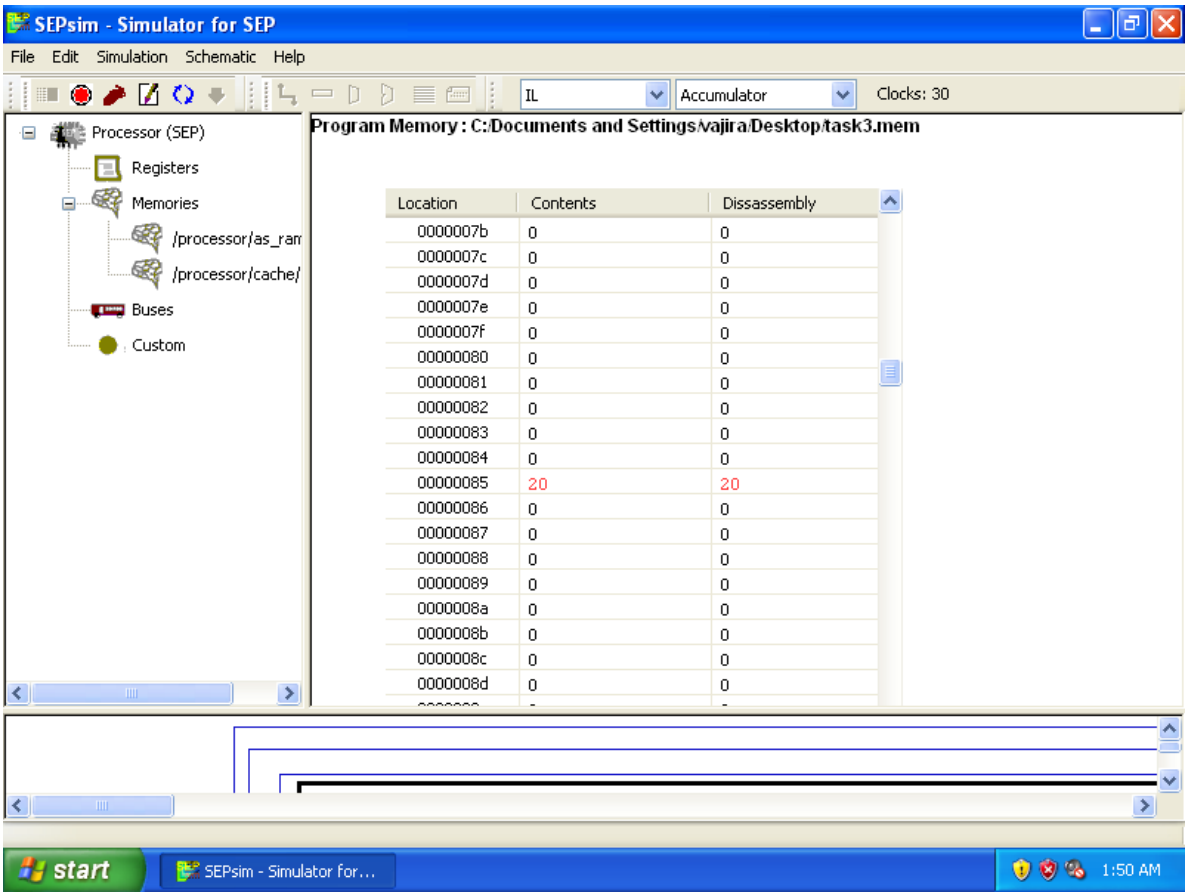
Here we using,
Inc – It will add +1 value to the accumulator / increment a value
nop – end the program.

TASK – 03


C:\Documents and Settings\vajira\Desktop\task3.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #15
add #5
storeacc 0x85
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
start
SEPsim - Simulator for...
6:37 PM

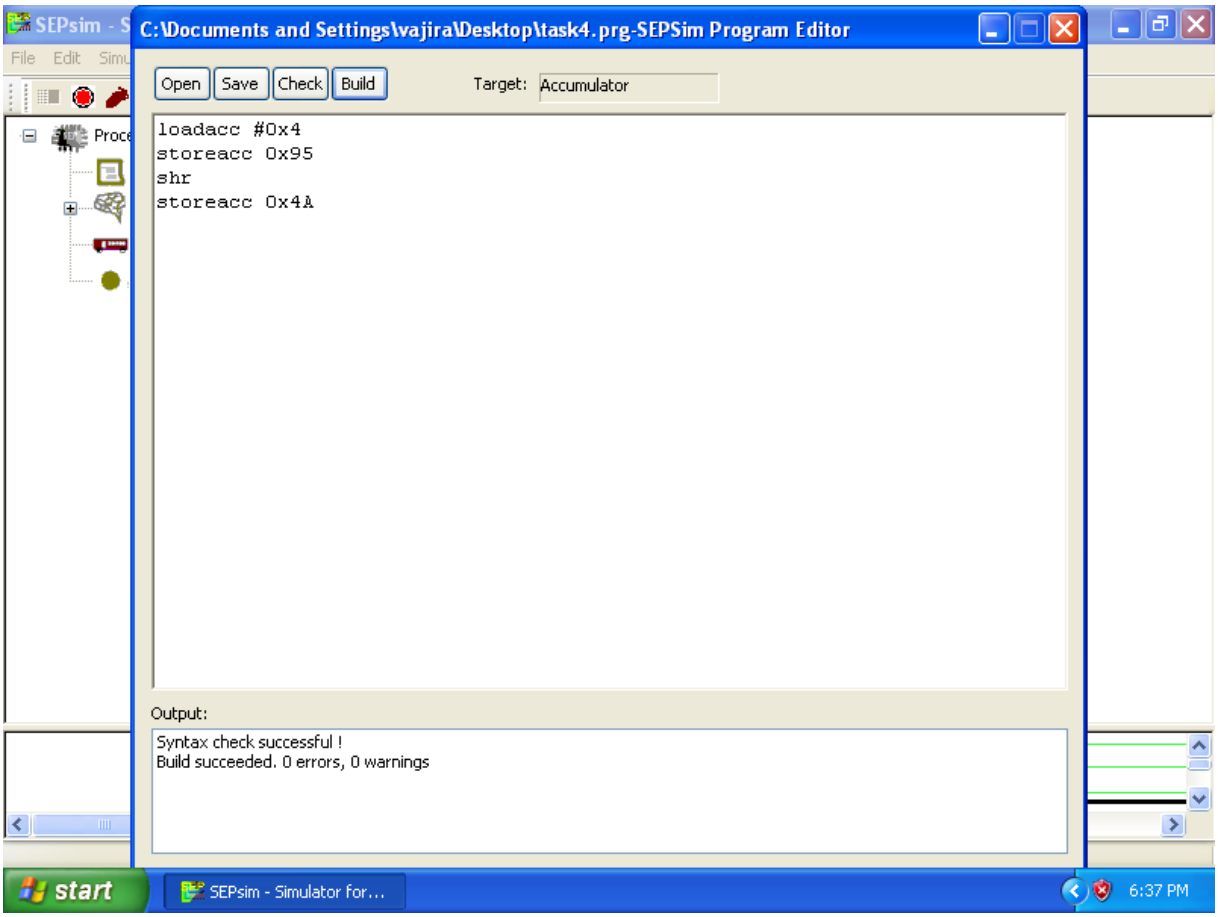

SEPsim - Simulator for SEP
File
Edit
Simulation
Schematic
Help
RTL
Accumulator
Clocks: 0
Processor (SEP)
Registers
Memories
Buses
Custom
Program Memory : C:\Documents and Settings\vajira\Desktop\task3.mem
Location
Contents
Dissassembly
00000000 63488 loadacc # 15
00000001 480
00000002 0
00000003 32768 add # 5
00000004 160
00000005 0
00000006 13312 storeacc def ..
00000007 4256
00000008 0
00000009 0 0
0000000a 0 0
0000000b 0 0
0000000c 0 0
0000000d 0 0
0000000e 0 0
0000000f 0 0
00000010 0 0
00000011 0 0
00000012 0 0
start
SEPsim - Simulator for...
1:46 AM

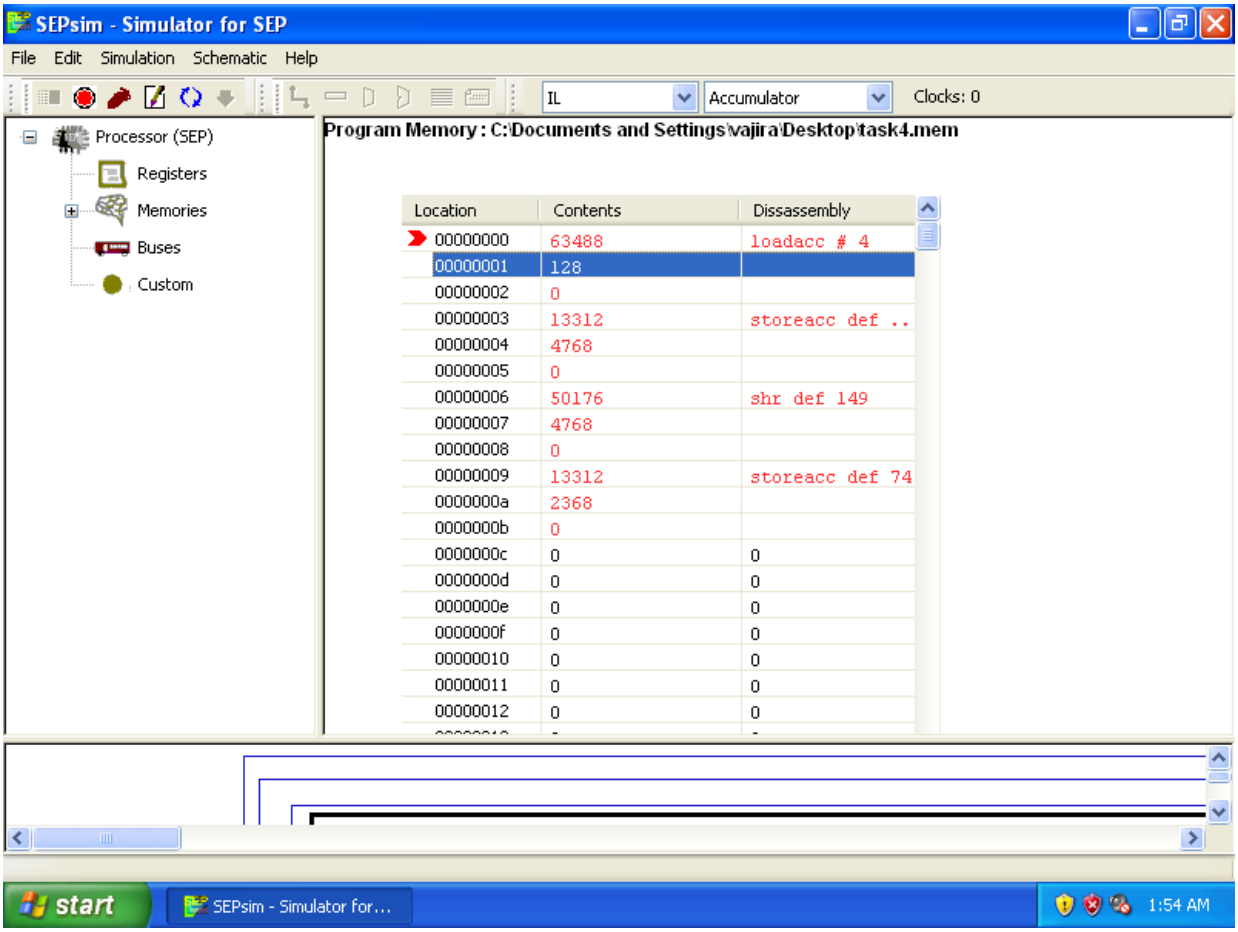


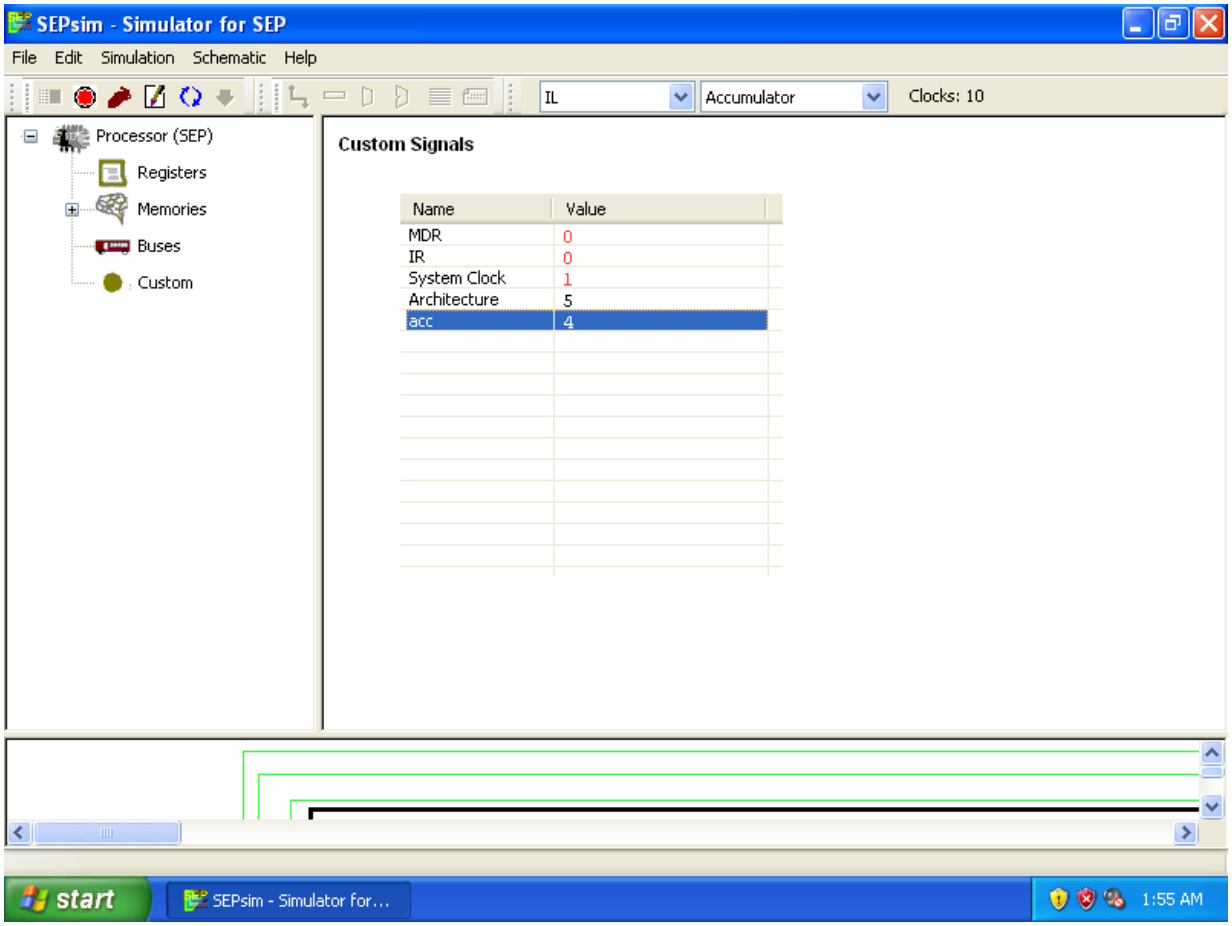


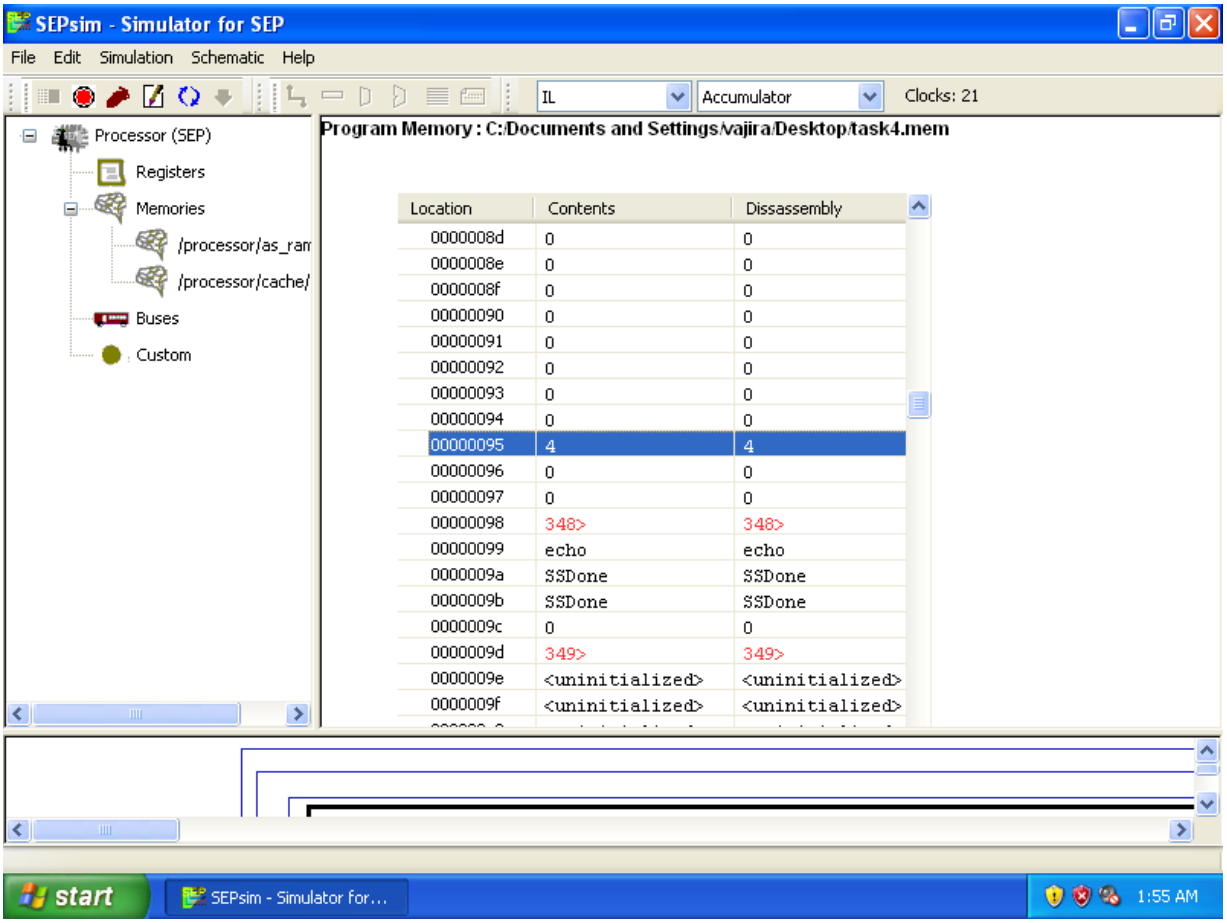
Load value 15 into accumulator, then add 5 and store accumulator value in 0x85 location.

TASK - 04

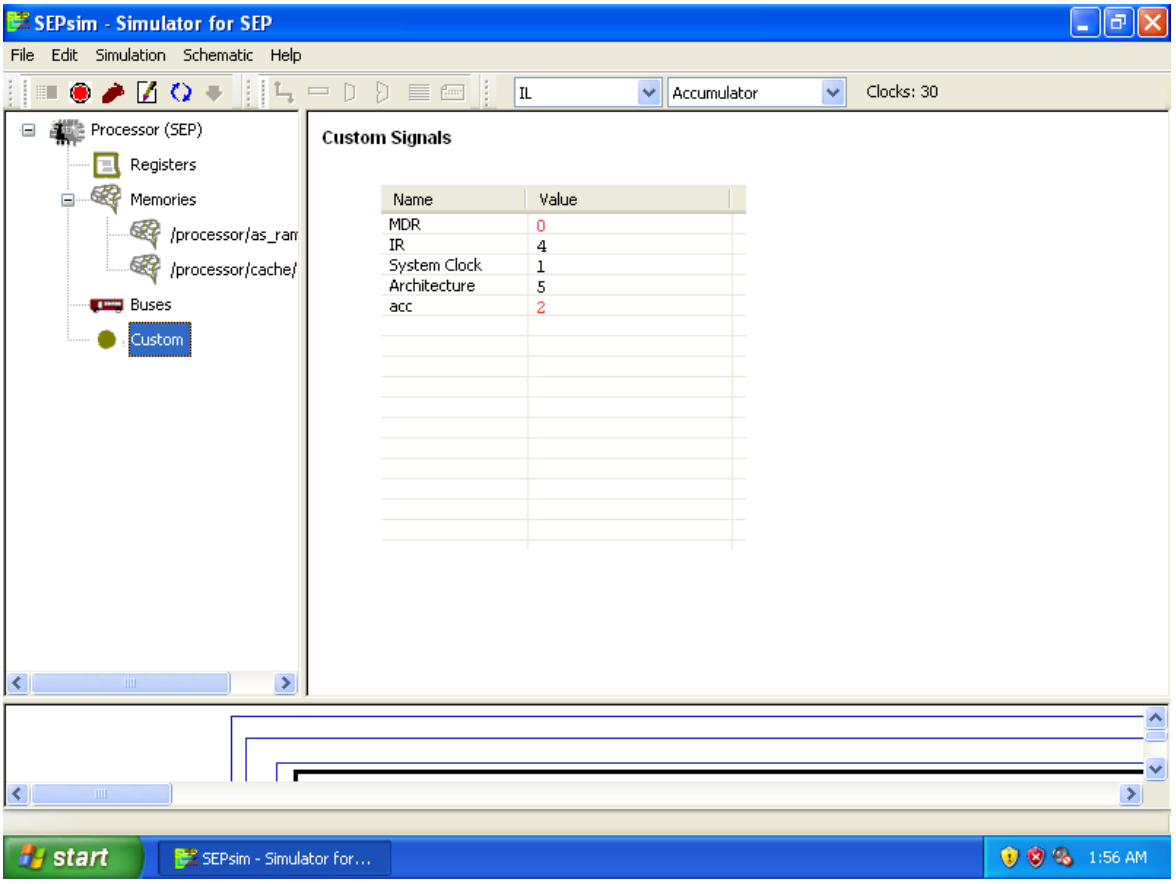

C:\Documents and Settings\vajira\Desktop\task4.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #0x4
storeacc 0x95
shr
storeacc 0x4A
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
start
SEPsim - Simulator for...
6:37 PM

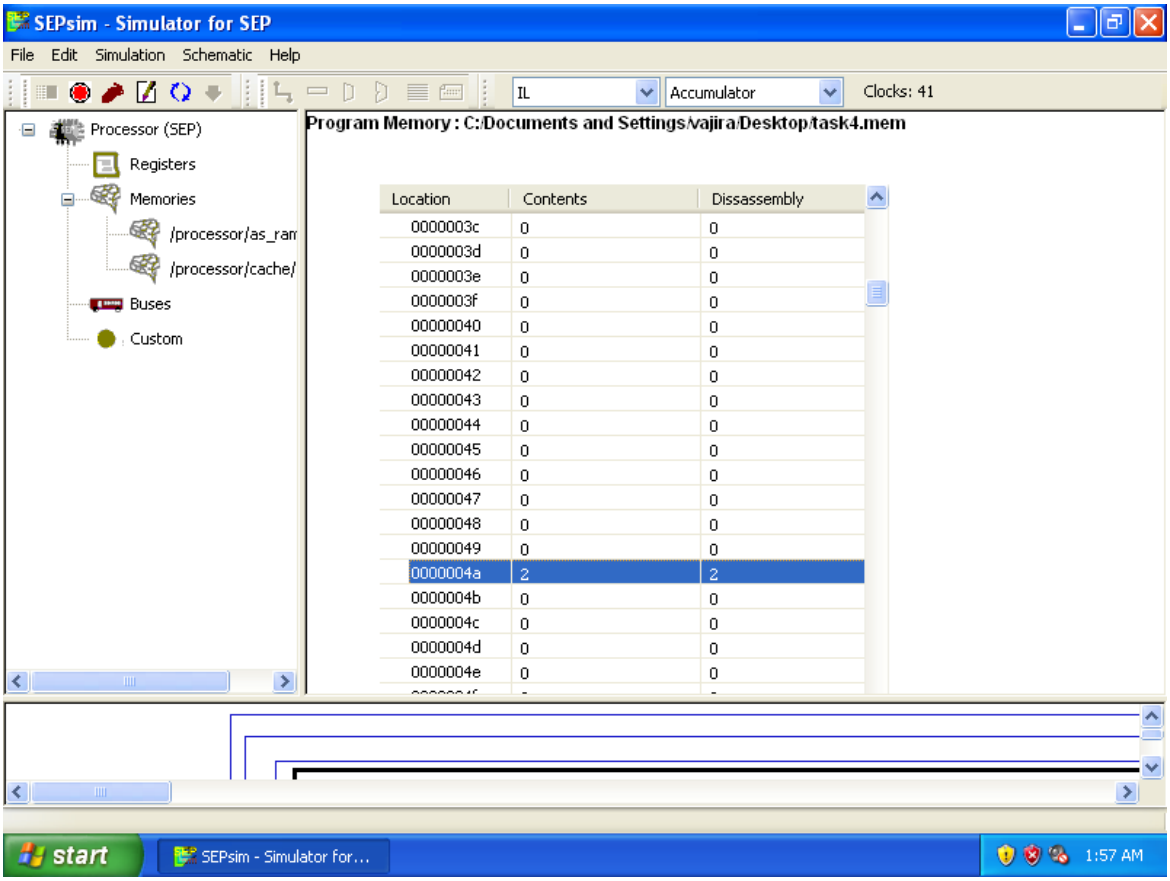

SEPsim - Simulator for SEP
File
Edit
Simulation
Schematic
Help
IL
Accumulator
Clocks: 0
Processor (SEP)
Registers
Memories
Buses
Custom
Program Memory : C:\Documents and Settings\vajira\Desktop\task4.mem
Location
Contents
Dissassembly
00000000 63488 loadacc # 4
00000001 128
00000002 0
00000003 13312 storeacc def ..
00000004 4768
00000005 0
00000006 50176 shr def 149
00000007 4768
00000008 0
00000009 13312 storeacc def 74
0000000a 2368
0000000b 0
0000000c 0 0
0000000d 0 0
0000000e 0 0
0000000f 0 0
00000010 0 0
00000011 0 0
00000012 0 0
start
SEPsim - Simulator for...
1:54 AM


SEPsim - Simulator for SEP
File Edit Simulation Schematic Help
IL
Accumulator
Clocks: 10
Processor (SEP)
Registers
Memories
Buses
Custom
Custom Signals
Name
Value
MDR
0
IR
0
System Clock
1
Architecture
5
acc
4
start
SEPsim - Simulator for...
1:55 AM

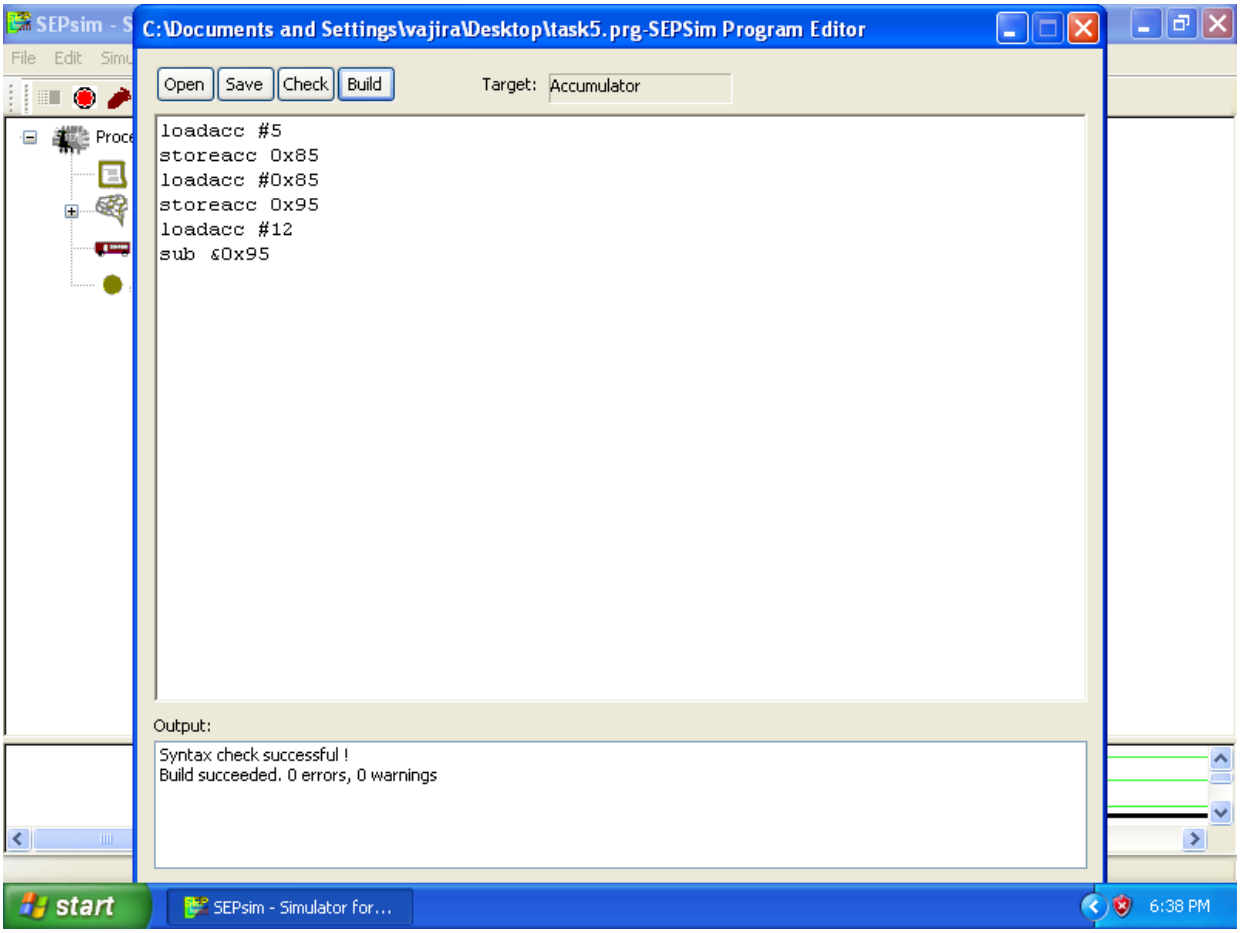

SEPsim - Simulator for SEP
File Edit Simulation Schematic Help
IL
Accumulator
Clocks: 21
Processor (SEP)
Registers
Memories
/processor/as_ram
/processor/cache/
Buses
Custom
Program Memory : C:/Documents and Settings/vajira/Desktop/task4.mem
Location
Contents
Disassembly
0000008d 0 0
0000008e 0 0
0000008f 0 0
00000090 0 0
00000091 0 0
00000092 0 0
00000093 0 0
00000094 0 0
00000095 4 4
00000096 0 0
00000097 0 0
00000098 348> 348>
00000099 echo echo
0000009a SSDone SSDone
0000009b SSDone SSDone
0000009c 0 0
0000009d 349> 349>
0000009e <uninitialized> <uninitialized>
0000009f <uninitialized> <uninitialized>
start
SEPsim - Simulator for...
1:55 AM

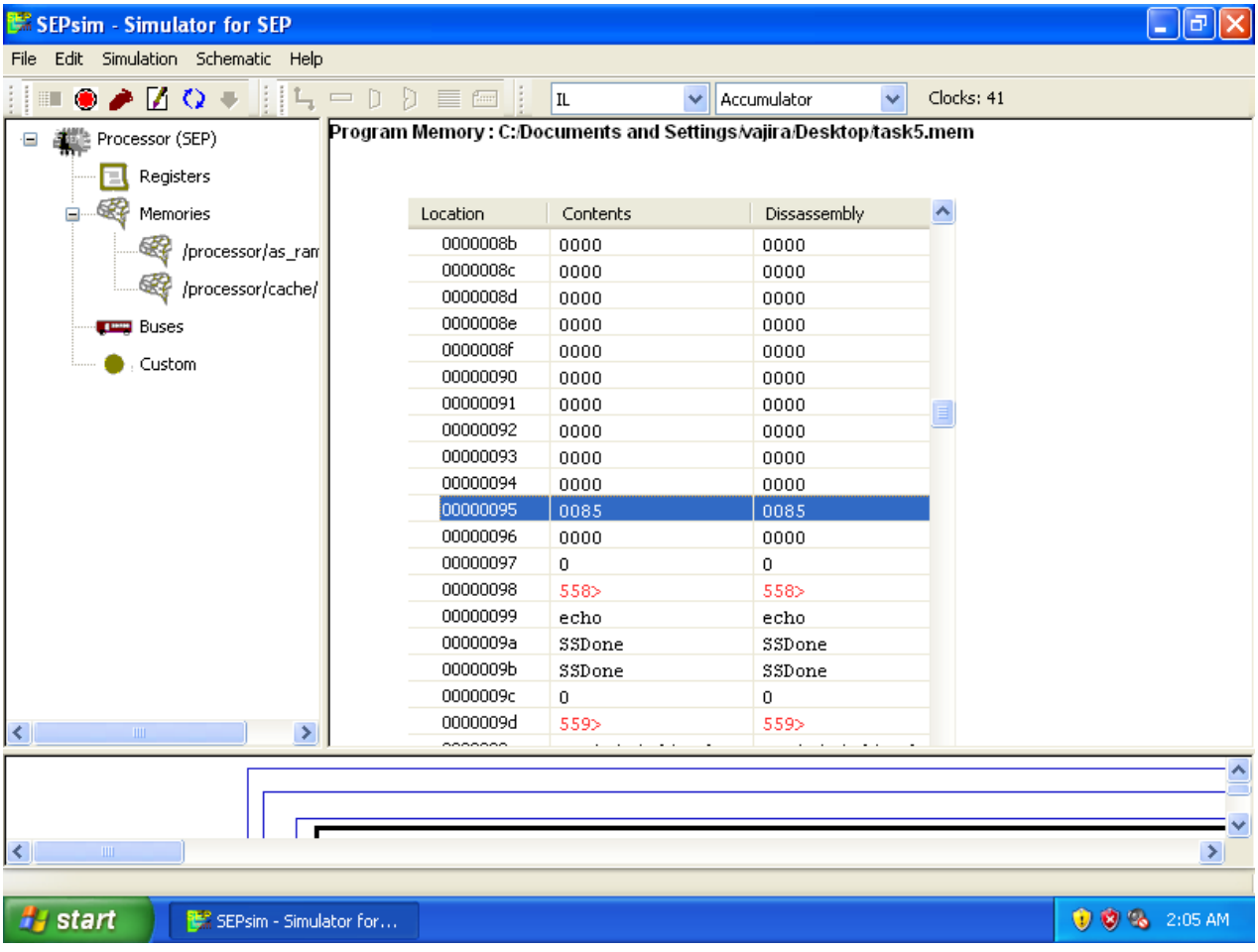
Load hexadecimal value 4 into accumulator, then store the value into 0x95 location,
Shr – Shift Right that value by 1 bit, after that it will come value 2 as hexadecimal and store it in the 0x4a location.



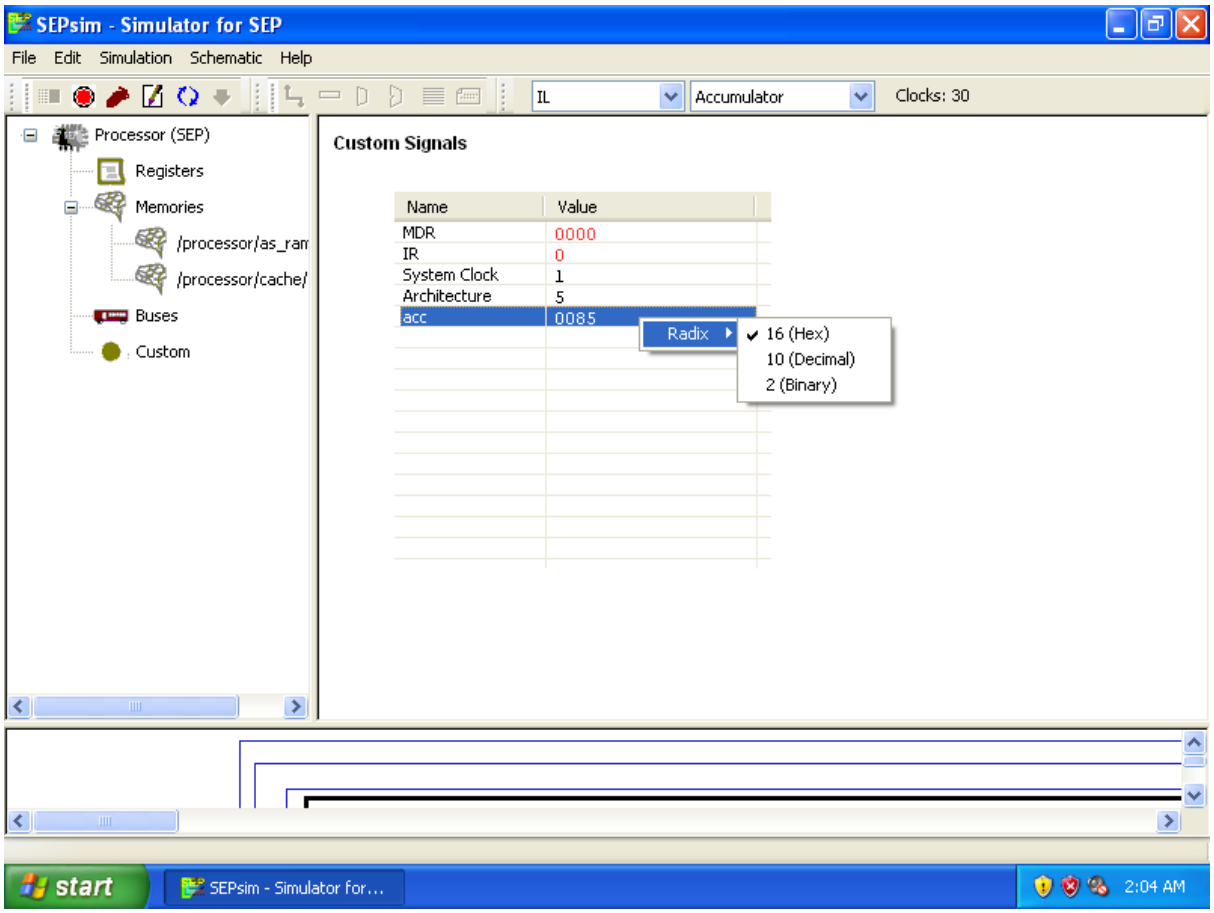


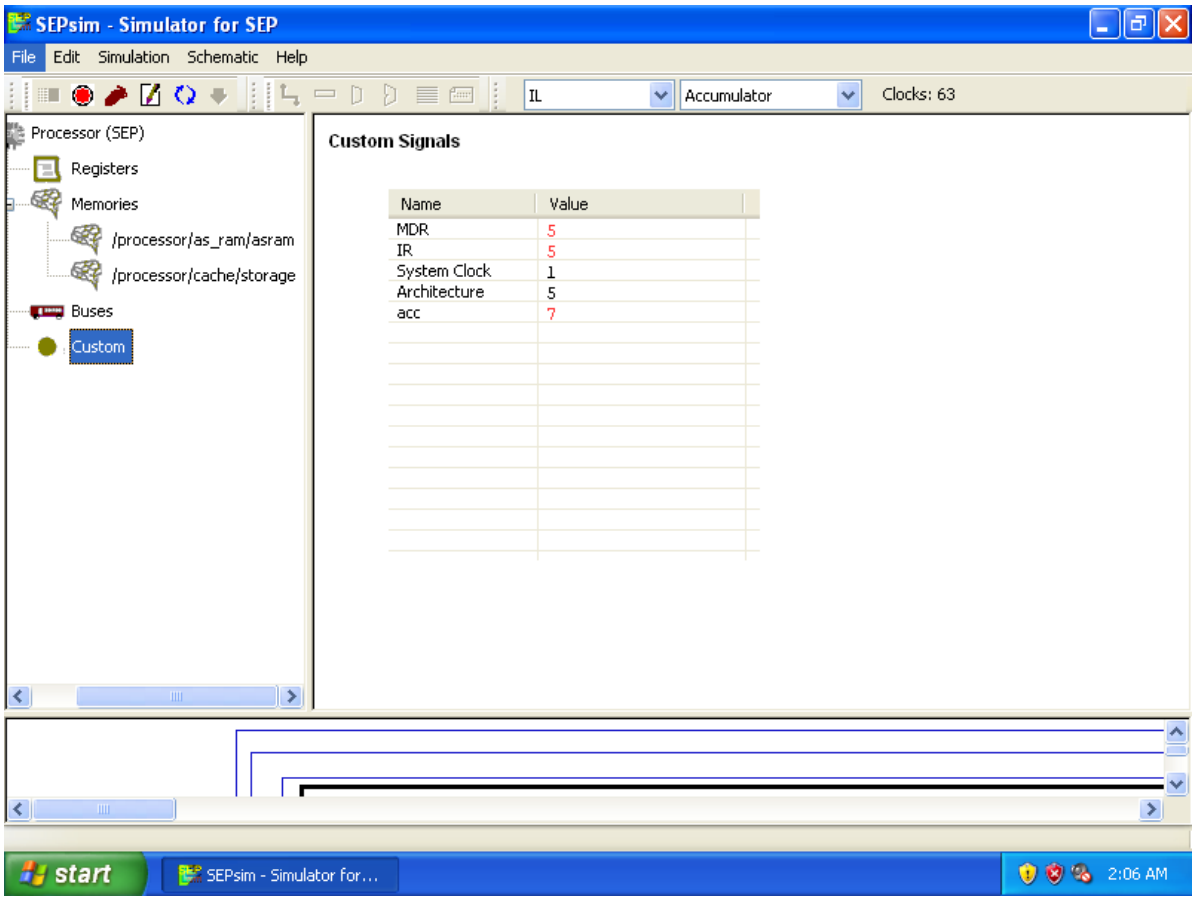
TASK – 05


C:\Documents and Settings\vajira\Desktop\task5.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #5
storeacc 0x85
loadacc #0x85
storeacc 0x95
loadacc #12
sub &0x95
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
start
SEPsim - Simulator for...
6:38 PM

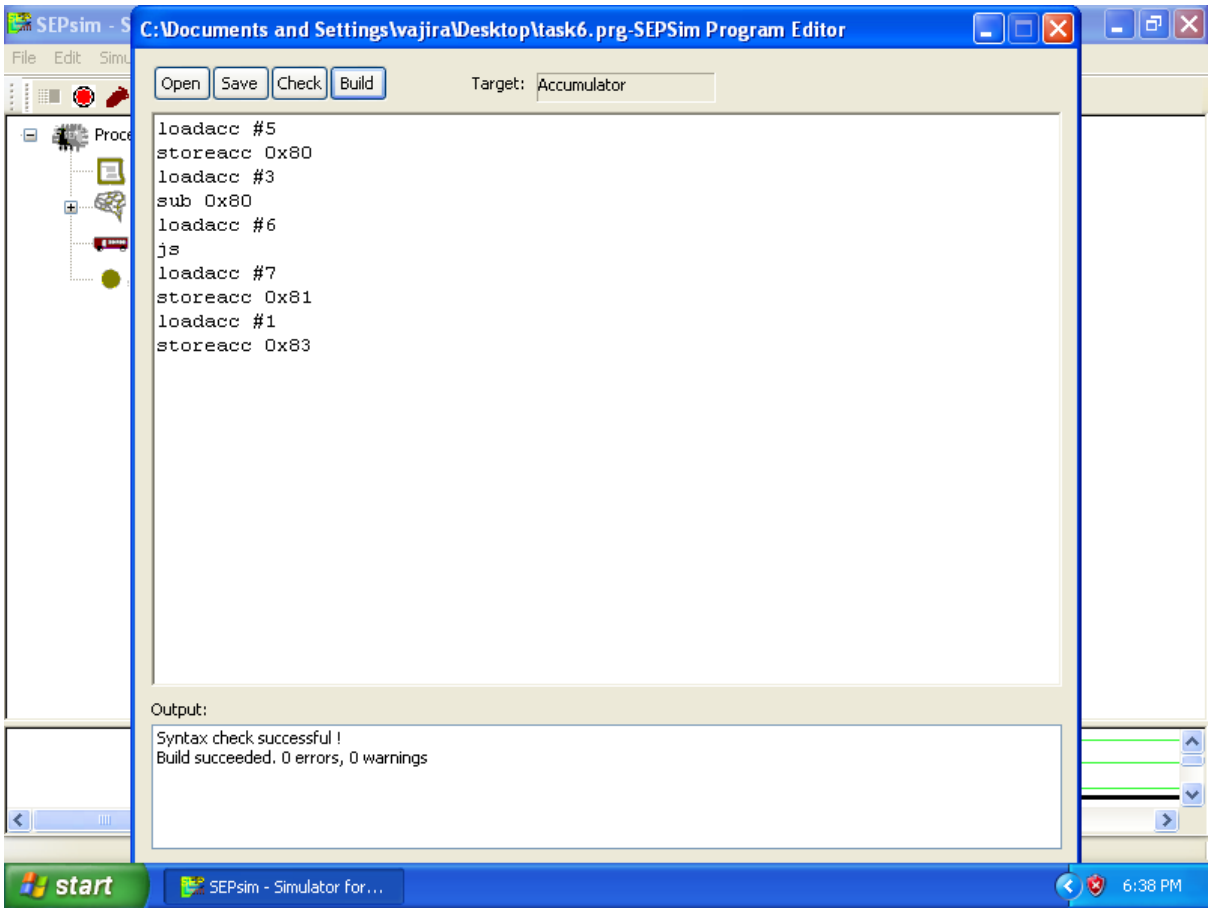

SEPsim - Simulator for SEP
File Edit Simulation Schematic Help
IL
Accumulator
Clocks: 41
Processor (SEP)
Registers
Memories
/processor/as_ram
/processor/cache/
Buses
Custom
Program Memory : C:/Documents and Settings/vajira/Desktop/task5.mem
Location Contents Dissassembly
0000008b 0000 0000
0000008c 0000 0000
0000008d 0000 0000
0000008e 0000 0000
0000008f 0000 0000
00000090 0000 0000
00000091 0000 0000
00000092 0000 0000
00000093 0000 0000
00000094 0000 0000
00000095 0085 0085
00000096 0000 0000
00000097 0 0
00000098 558> 558>
00000099 echo echo
0000009a SSDone SSDone
0000009b SSDone SSDone
0000009c 0 0
0000009d 559> 559>
start
SEPsim - Simulator for...
2:05 AM

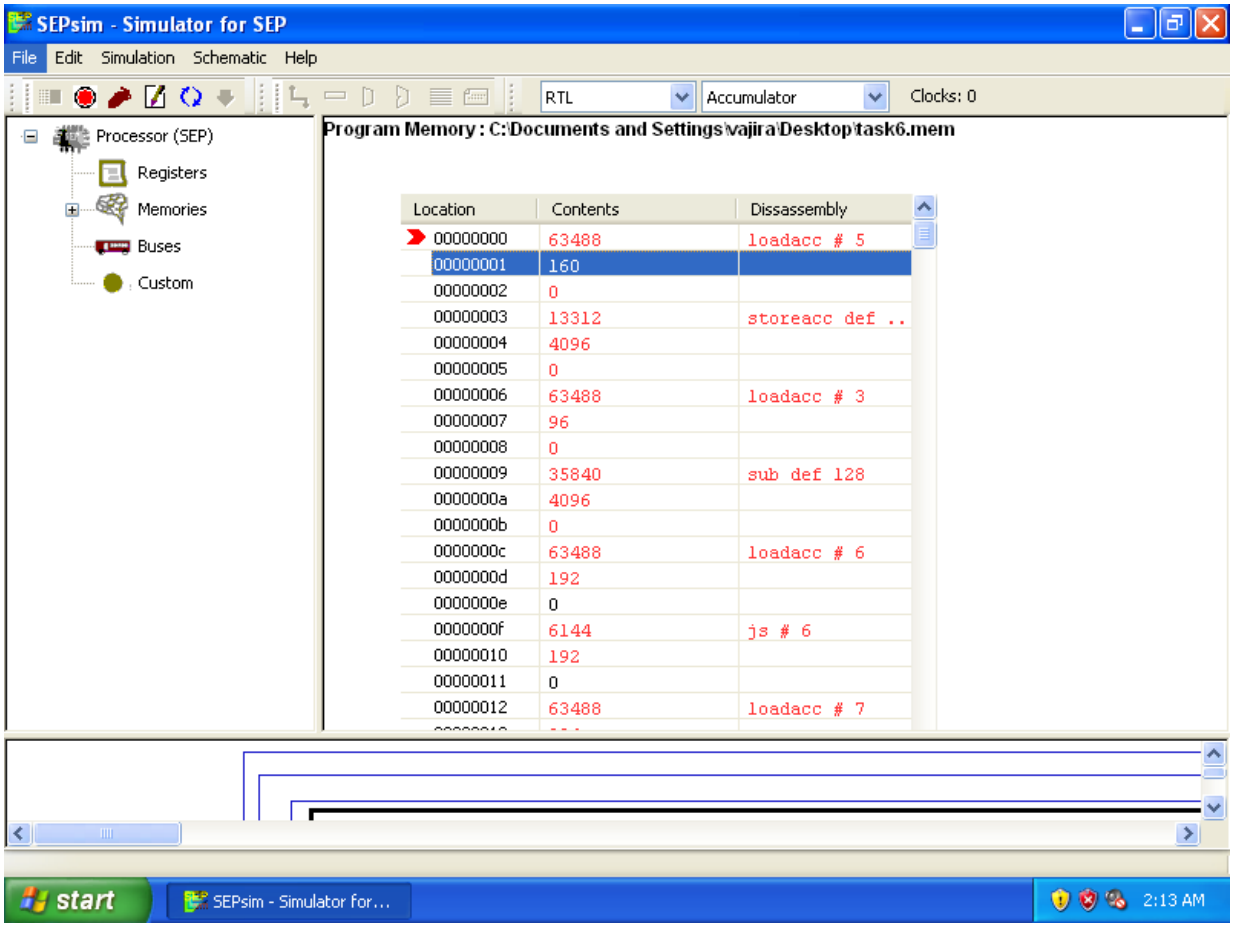
Load value 5 in accumulator, then store that value in 0x85 location, after that load 0x85 in to accumulator, then store that value in 0x95. This is indirect way to load number 5 in to 0x95 location,
After that load integer 12, then using sub opcode and subtract the location 0x95. Finally, the answer will be 7.



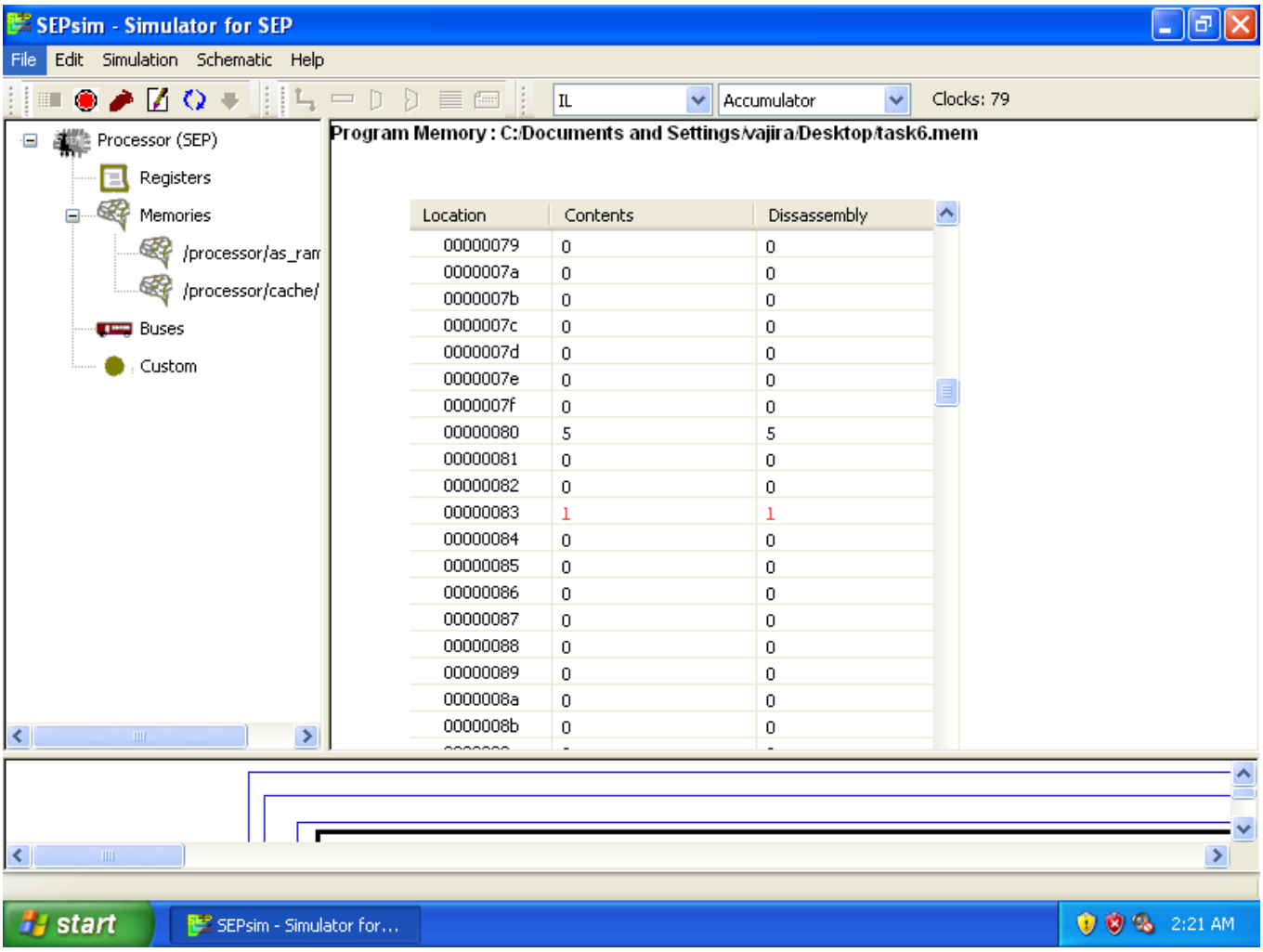


TASK – 06

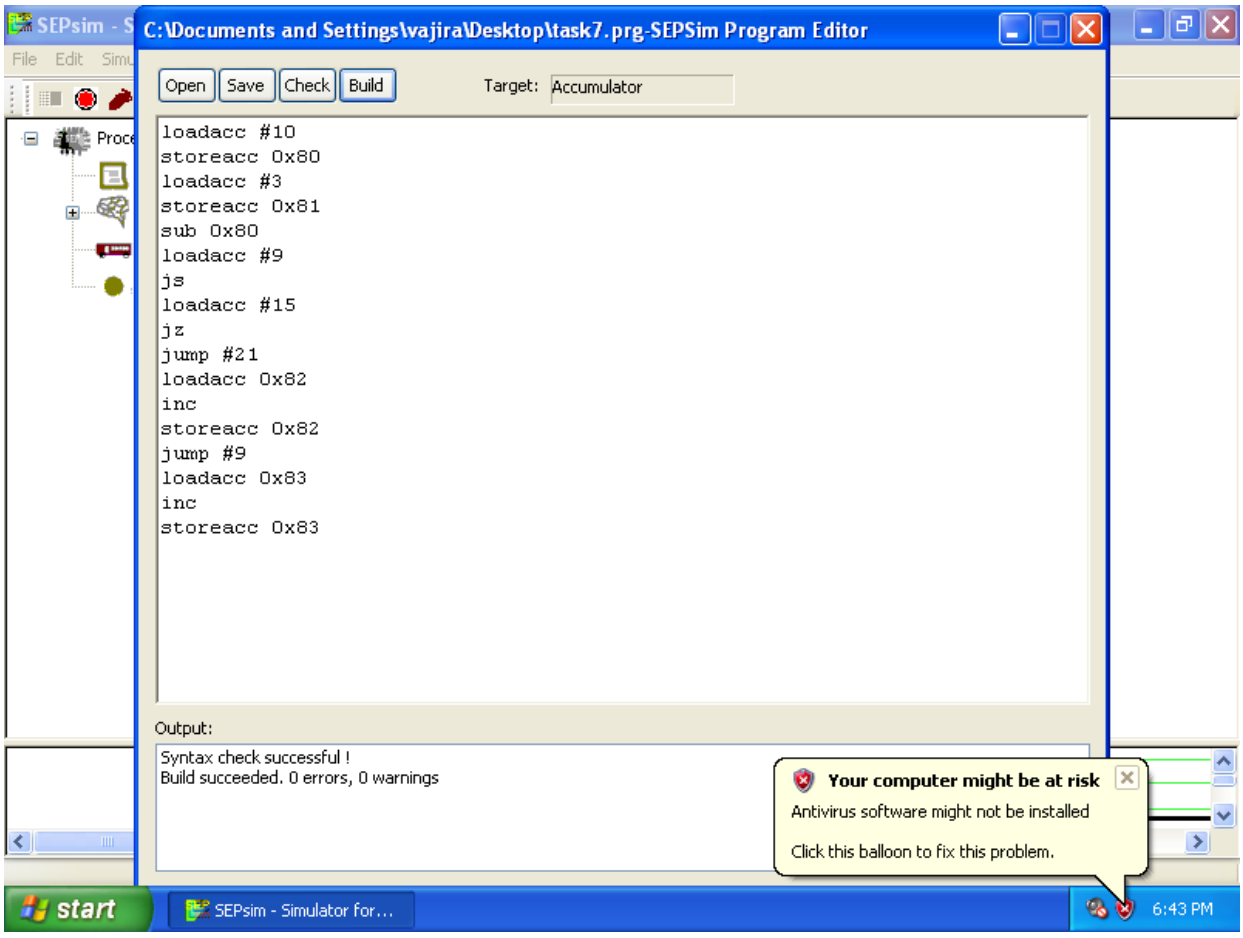

C:\Documents and Settings\vajira\Desktop\task6.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #5
storeacc 0x80
loadacc #3
sub 0x80
loadacc #6
js
loadacc #7
storeacc 0x81
loadacc #1
storeacc 0x83
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
start
SEPsim - Simulator for...
6:38 PM

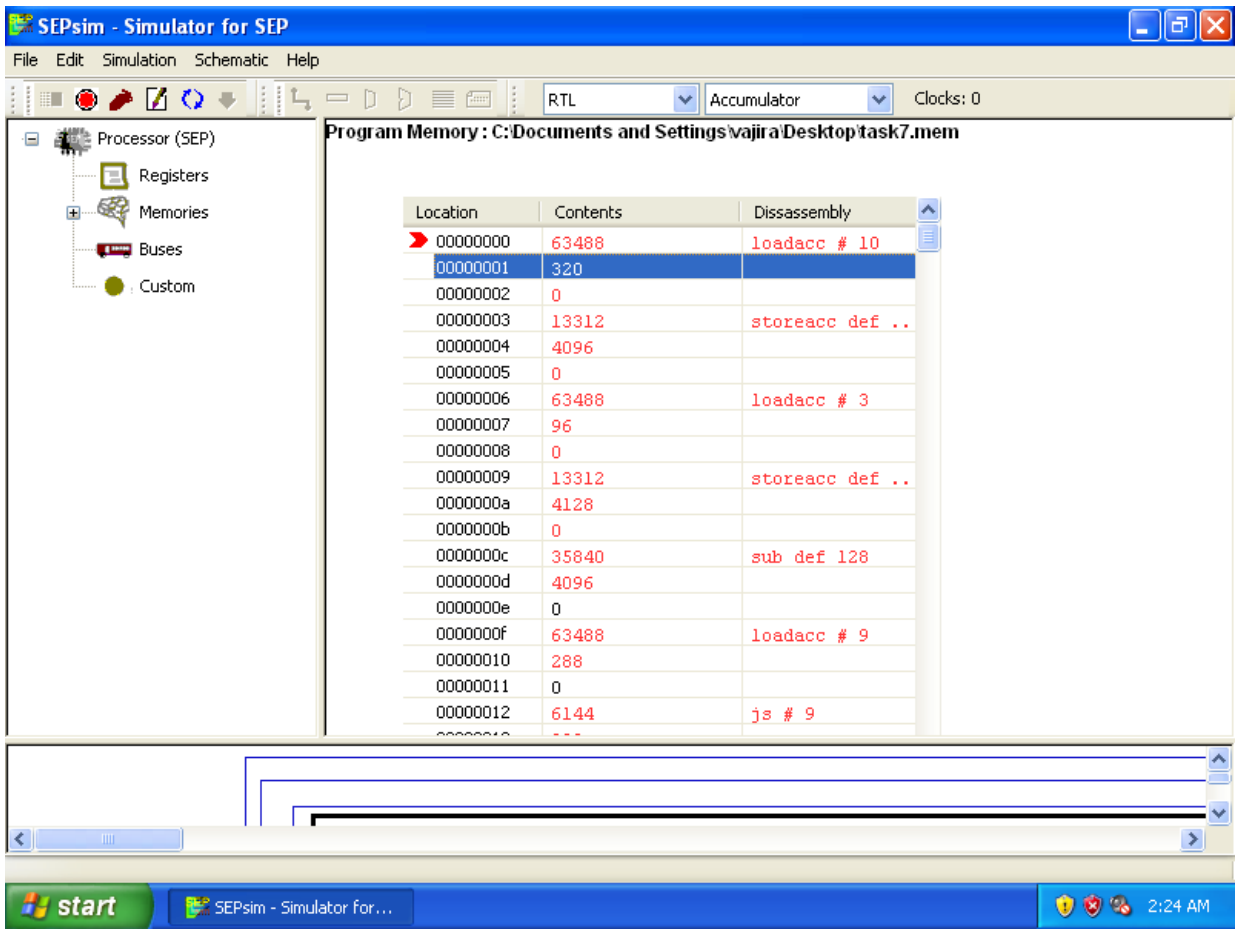

SEPsim - Simulator for SEP
File
Edit
Simulation
Schematic
Help
RTL
Accumulator
Clocks: 0
Processor (SEP)
Registers
Memories
Buses
Custom
Program Memory : C:\Documents and Settings\vajira\Desktop\task6.mem
Location
Contents
Disassembly
00000000 63488 loadacc # 5
00000001 160
00000002 0
00000003 13312 storeacc def ..
00000004 4096
00000005 0
00000006 63488 loadacc # 3
00000007 96
00000008 0
00000009 35840 sub def 128
0000000a 4096
0000000b 0
0000000c 63488 loadacc # 6
0000000d 192
0000000e 0
0000000f 6144 js # 6
00000010 192
00000011 0
00000012 63488 loadacc # 7
start
SEPsim - Simulator for...
2:13 AM

Here this task consists how to use jumpsign , we load 5 into accumulator and store the value in 0x80 location, then we load 3 and using sub opcode subtract the value 0x80, here we can see the accumulator value will be -2 so if the accumulator value is in negative jumpsign will work and avoid some steps by the simulator and it will load number 1 and stored in 0x83.



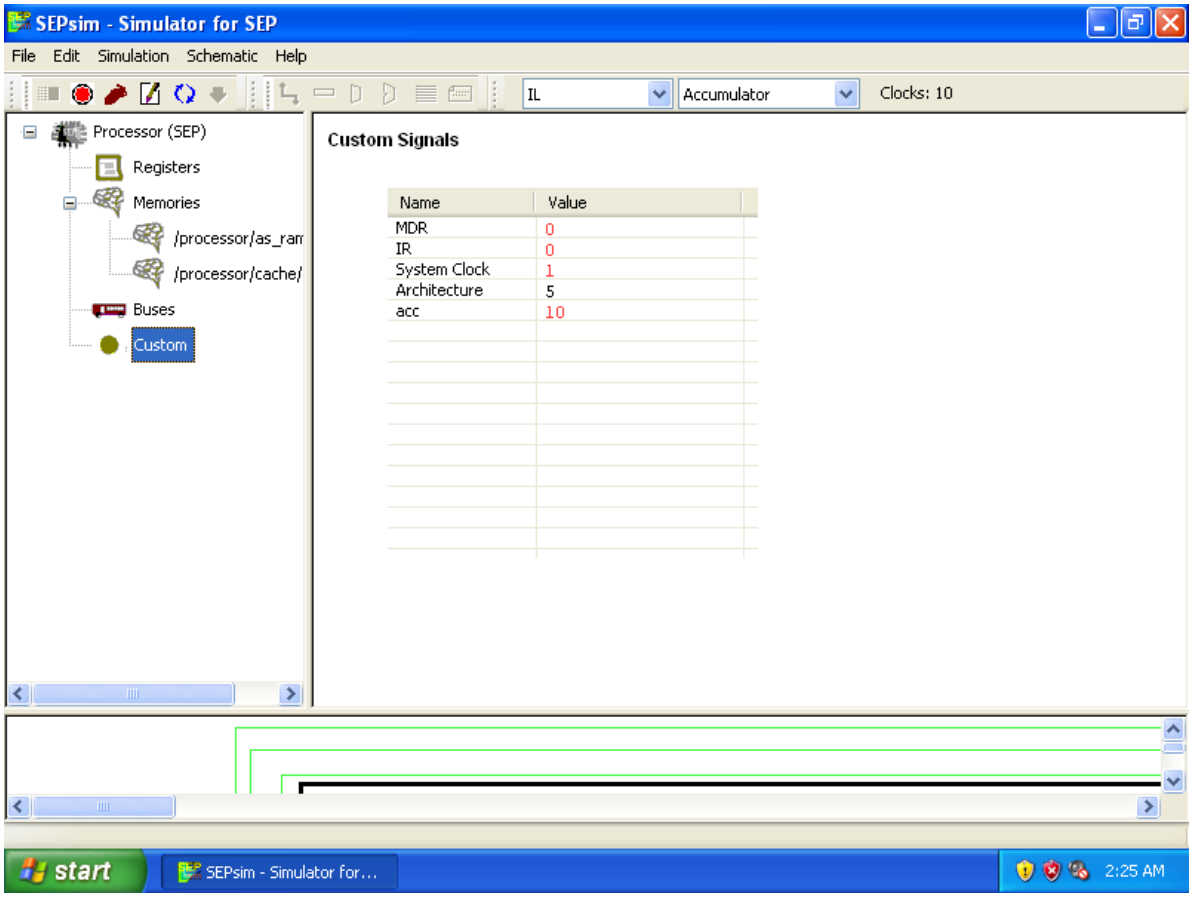
TASK – 07

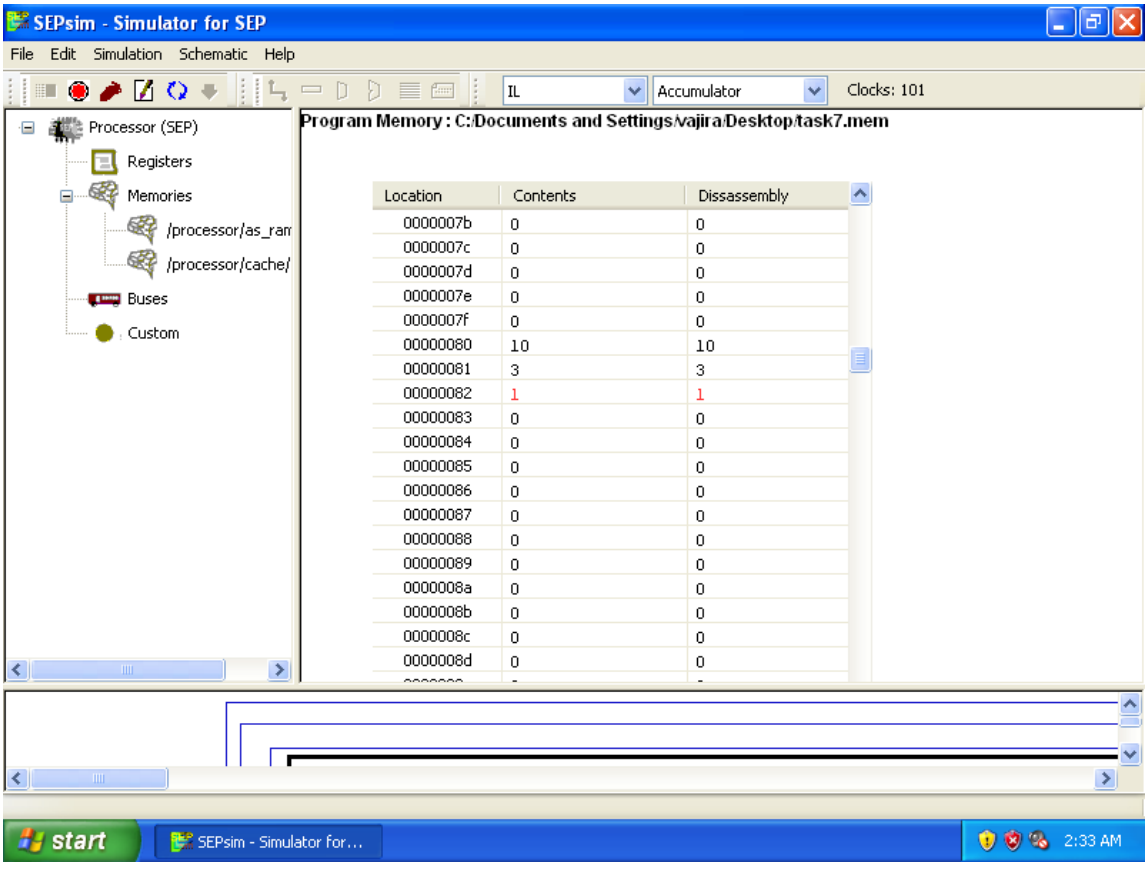

C:\Documents and Settings\vajira\Desktop\task7.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #10
storeacc 0x80
loadacc #3
storeacc 0x81
sub 0x80
loadacc #9
js
loadacc #15
jz
jump #21
loadacc 0x82
inc
storeacc 0x82
jump #9
loadacc 0x83
inc
storeacc 0x83
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
Your computer might be at risk
Antivirus software might not be installed
Click this balloon to fix this problem.
start
SEPsim - Simulator for...
6:43 PM


SEPsim - Simulator for SEP
File Edit Simulation Schematic Help
RTL
Accumulator
Clocks: 0
Processor (SEP)
Registers
Memories
Buses
Custom
Program Memory : C:\Documents and Settings\vajira\Desktop\task7.mem
Location Contents Dissassembly
00000000 63488 loadacc # 10
00000001 320
00000002 0
00000003 13312 storeacc def ..
00000004 4096
00000005 0
00000006 63488 loadacc # 3
00000007 96
00000008 0
00000009 13312 storeacc def ..
0000000a 4128
0000000b 0
0000000c 35840 sub def 128
0000000d 4096
0000000e 0
0000000f 63488 loadacc # 9
00000010 288
00000011 0
00000012 6144 js # 9
start
SEPsim - Simulator for...
2:24 AM

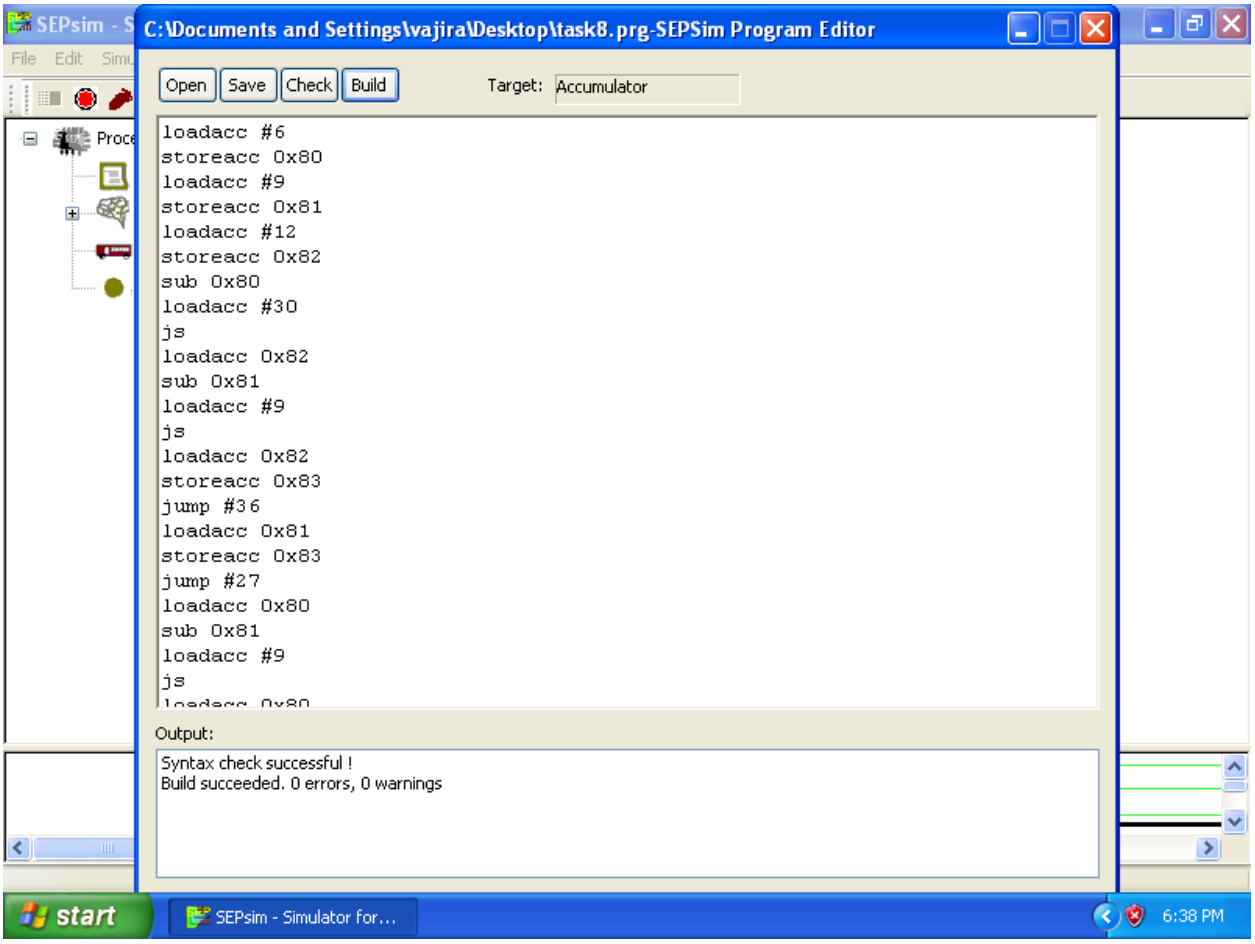
Here in task 7, we declare a and b and stored in a define location, and if the subtraction value is negative the jumpSign will work, and a will be the largest no. and it will locate the value in 0x82.

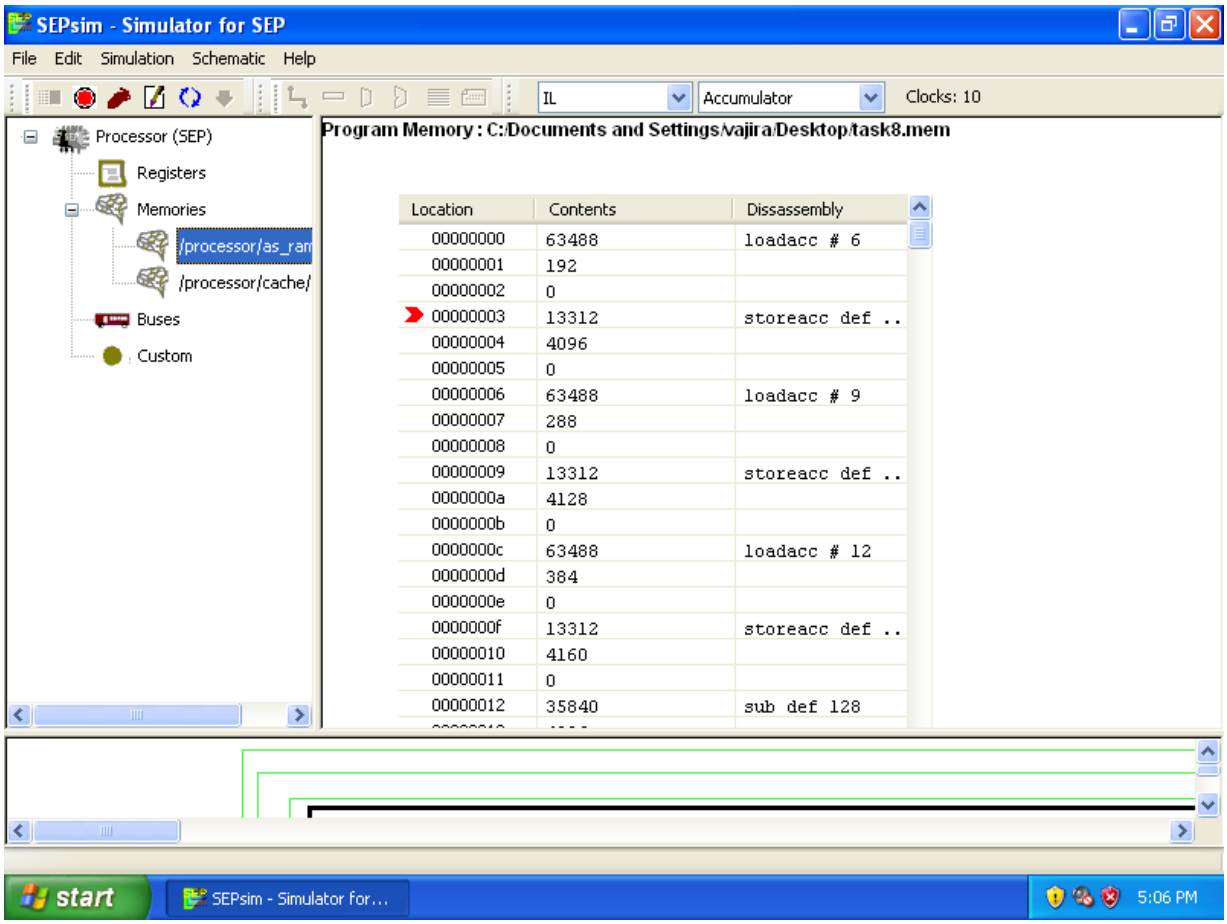
Otherwise jumpZero will work and locate the value in 0x83.

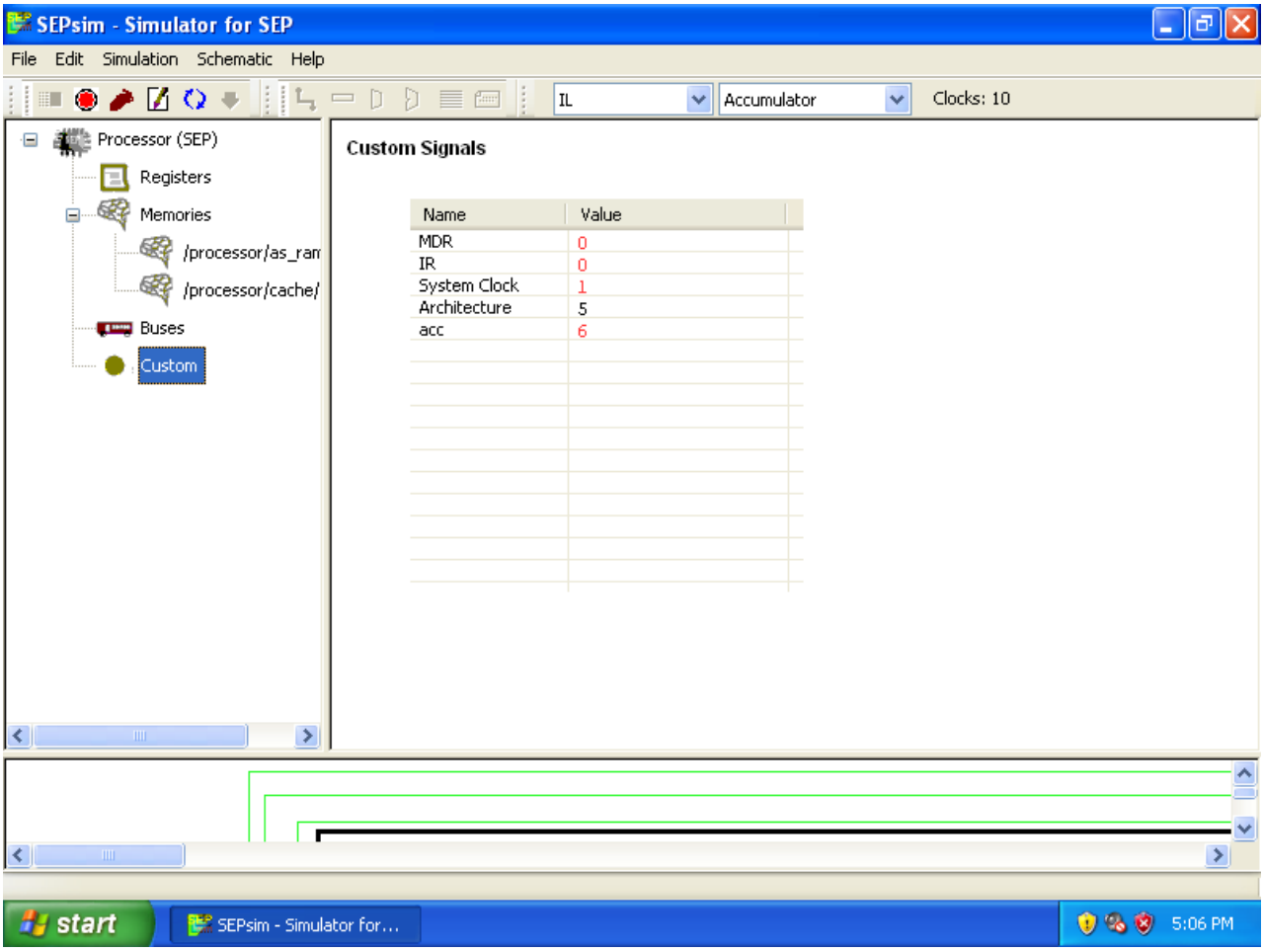


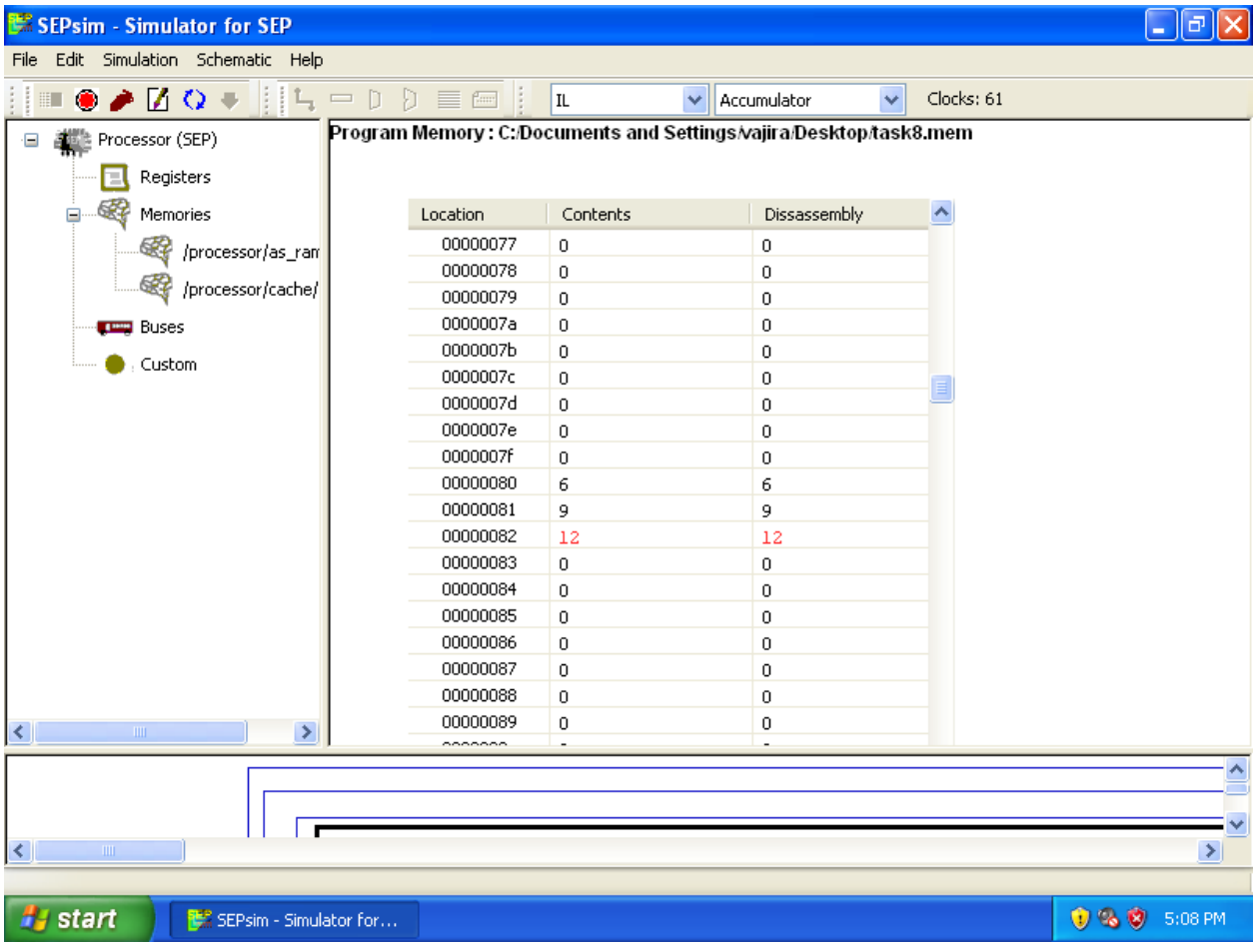


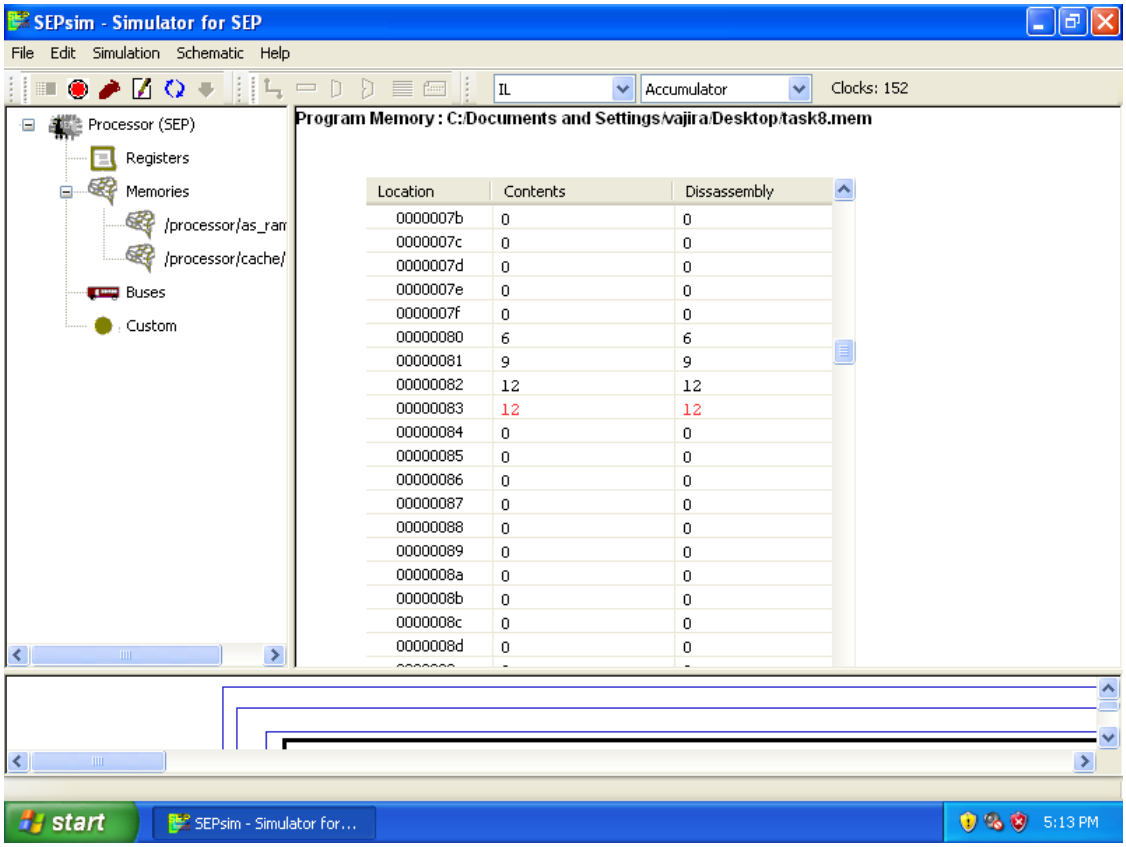
TASK - 08


C:\Documents and Settings\vajira\Desktop\task8.prg-SEPSim Program Editor
Open
Save
Check
Build
Target:
Accumulator
loadacc #6
storeacc 0x80
loadacc #9
storeacc 0x81
loadacc #12
storeacc 0x82
sub 0x80
loadacc #30
js
loadacc 0x82
sub 0x81
loadacc #9
js
loadacc 0x82
storeacc 0x83
jump #36
loadacc 0x81
storeacc 0x83
jump #27
loadacc 0x80
sub 0x81
loadacc #9
js
Output:
Syntax check successful !
Build succeeded. 0 errors, 0 warnings
start
SEPsim - Simulator for...
6:38 PM


SEPsim - Simulator for SEP
File
Edit
Simulation
Schematic
Help
IL
Accumulator
Clocks: 10
Processor (SEP)
Registers
Memories
/processor/as_ram
/processor/cache/
Buses
Custom
Program Memory : C:/Documents and Settings/vajira/Desktop/task8.mem
Location
Contents
Disassembly
00000000 63488 loadacc # 6
00000001 192
00000002 0
00000003 13312 storeacc def ..
00000004 4096
00000005 0
00000006 63488 loadacc # 9
00000007 288
00000008 0
00000009 13312 storeacc def ..
0000000a 4128
0000000b 0
0000000c 63488 loadacc # 12
0000000d 384
0000000e 0
0000000f 13312 storeacc def ..
00000010 4160
00000011 0
00000012 35840 sub def 128
start
SEPsim - Simulator for...
5:06 PM

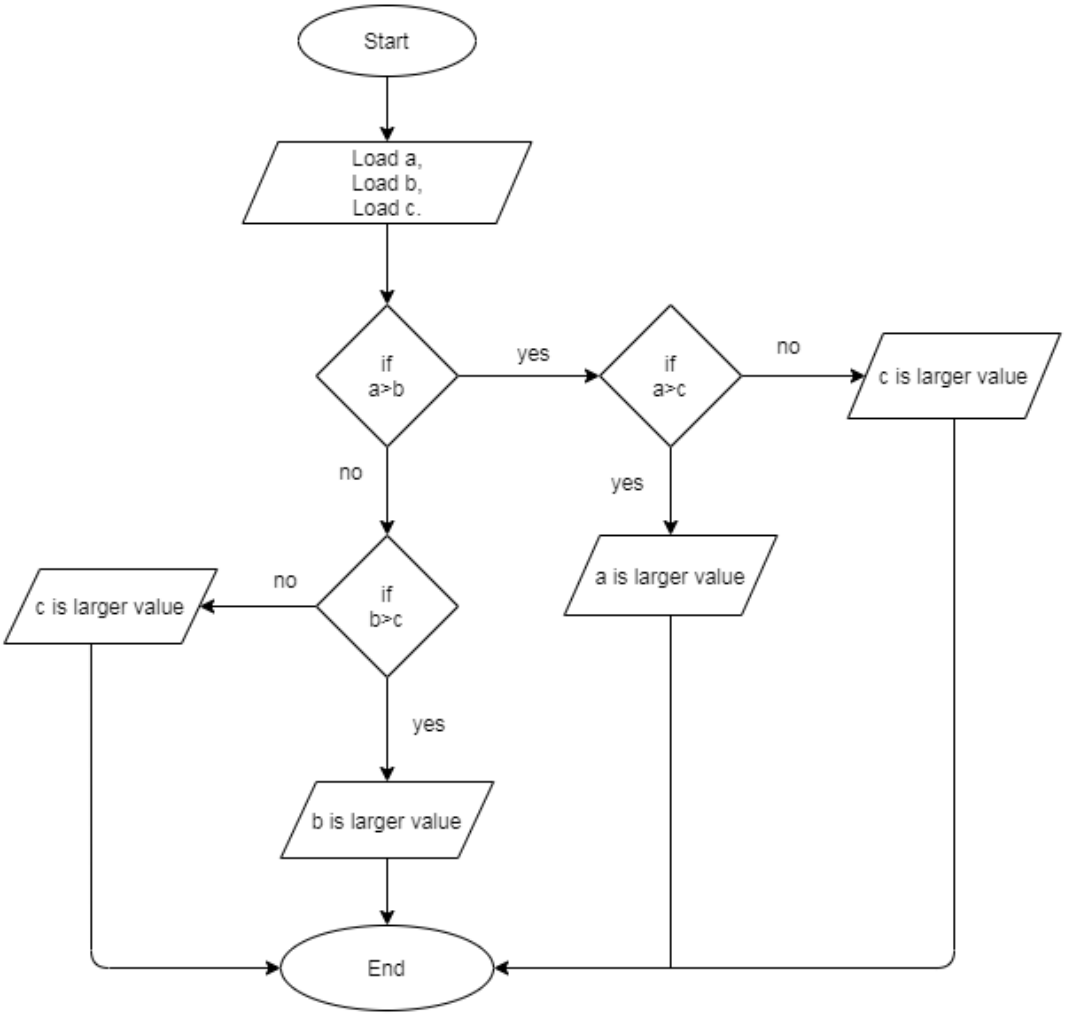

SEPsim - Simulator for SEP
File Edit Simulation Schematic Help
IL
Accumulator
Clocks: 10
Processor (SEP)
Registers
Memories
/processor/as_ram
/processor/cache/
Buses
Custom
Custom Signals
Name
Value
MDR 0
IR 0
System Clock 1
Architecture 5
acc 6
start
SEPsim - Simulator for...
5:06 PM


SEPsim - Simulator for SEP
File Edit Simulation Schematic Help
IL
Accumulator
Clocks: 61
Processor (SEP)
Registers
Memories
/processor/as_ram
/processor/cache/
Buses
Custom
Program Memory : C:/Documents and Settings/vajira/Desktop/task8.mem
Location
Contents
Disassembly
00000077 0 0
00000078 0 0
00000079 0 0
0000007a 0 0
0000007b 0 0
0000007c 0 0
0000007d 0 0
0000007e 0 0
0000007f 0 0
00000080 6 6
00000081 9 9
00000082 12 12
00000083 0 0
00000084 0 0
00000085 0 0
00000086 0 0
00000087 0 0
00000088 0 0
00000089 0 0
start
SEPsim - Simulator for...
5:08 PM



Here in task 8, we declare a, b and c, and stored in a define location and find the largest value using jumpSign concepts and locate the value 0x83.

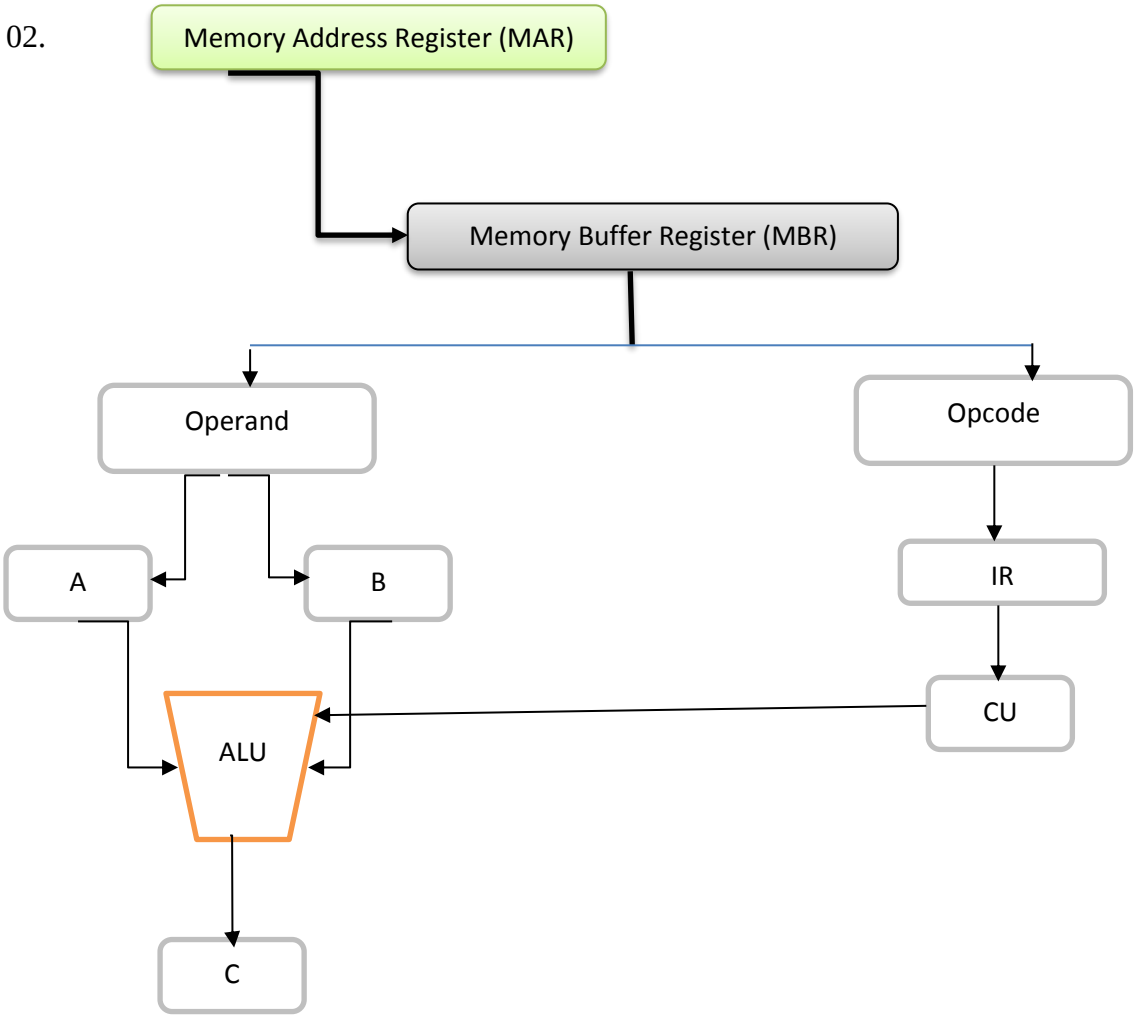
Ex. In (A-B) if (B>A) then Jump sign will calculate (C-B) if (B>C) then jumpSign will define (B) as larger number and print B then locate the B in 0x83.



Discussion Outcome

01. In Accumulator assembly we can divide in to 3 types, are
    A. Opcode
    B. Addressing Mode (Immediate, Direct, Indirect)
    C. Operand

02.



03. In this Hierarchy we can see, how the accumulator architecture will work. I understand Whole concepts via this architecture and how it's work.
04. I understand these three concepts and how it's worked.
    JS – JumpSign
    JZ – JumpZero
    OF – OverFlow

05. Problem Faced
    A. When I install the Virtual box it's not supported to my pc. After that I reinstall an old version of Virtual box and practice my work.
    B. I tried whole concepts a while more for understand it Through the practice. For the task7 and task 8 I tried to solve it. I hope it will satisfy the concepts.

S92064060