CPU-OS Simulator

Dr. Lucy J. Gudino
WILP & Department of CS & IS

BITS Pilani
Pilani Campus

Agenda

- Introduction to CPU OS Simulator
  - CPU Simulator
  - Compiler Simulator
  - OS Simulator
- Introduction to CPU Simulator
- Instruction set architecture
- Examples
- Question and Answer Session

Introduction

- Developed by Besim Mustafa, Department of Computing, Edge Hill University, U.K.
- Link to down load CPU-OS Simulator Version 7.5: http://www.teach-sim.com/
- Three components of CPU-OS simulator
  - The CPU Simulator
  - The Compiler Simulator
  - The OS Simulator

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]

CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T

Cache - Pipeline | Execution Unit

Pipeline
Single pipeline | Dual pipeline
Select pipeline: 0
SHOW PIPELINE...

Cache
Select cache type: Data
SHOW CACHE...

SPECIAL CPU REGISTERS

PC: 0
SR: 0
SP: 8096
BR: 0

SR Status Flag
OV Z N

CPU Mode
User
Kernel

IR
MAR: 0
MDR

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)	C	Val (D)
R06	0		
R07	0		
R08	0		
R09	0		
R10	0		
R11	0		
R12	0		
R13	0		
R14	0		
R15	0		
R16	0		
R17	0		
R18	0		
R19	0		
R20	0		
R21	0		
R22	0		
R23	0		
R24	0		
R25	0		
R26	0		
R27	0		
R28	0		
R29	0		
R30	0		
R31	0		

PROGRAM LIST

Name	Base	Start	Type

LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE

PROGRAM STACK (RAM)

	Pos	Val (D)	Addr

Program | Instructions | Optimize - Assemble

New Program
Program Name
Pages: 1
Base Address
ADD

Programs
SAVE...
Program List: ALL
LOAD...
Base Address: -1
COPY TO CLIPBOARD

Program Control | CPU View | CPU Help

STEP
by instruction
by single tick
RUN
Fast
Slow
RESET PROGRAM
STOP
SHOW PCB...

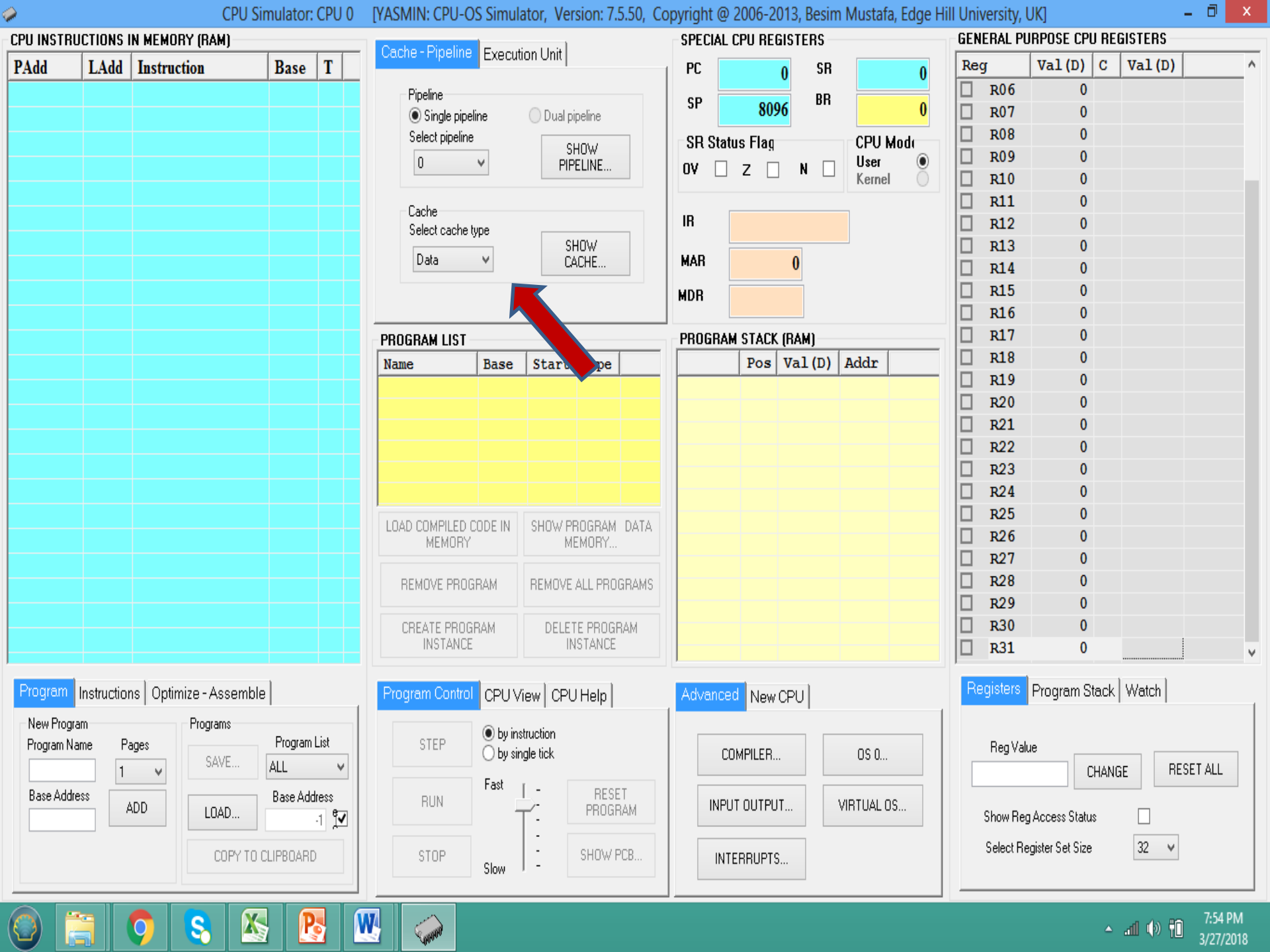
Advanced | New CPU

COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...

Registers | Program Stack | Watch

Reg Value
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size: 32

7:54 PM
3/27/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd
LAdd
Instruction
Base
T
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC
0
SR
0
SP
8096
BR
0
SR Status Flag
OV
Z
N
CPU Mode
User
Kernel
IR
MAR
0
MDR
GENERAL PURPOSE CPU REGISTERS
Reg
Val (D)
C
Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name
Base
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos
Val (D)
Addr
Program
Instructions
Optimize - Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
ALL
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
STEP
by instruction
by single tick
RUN
Fast
Slow
STOP
RESET PROGRAM
SHOW PCB...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
7:54 PM
3/27/2018

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]

CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction

SPECIAL CPU REGISTERS

PC: 0 SR: 0

SP: 8096 BR: 0

SR Status Flag

OV Z N

CPU Mode

User

Kernel

PROGRAM STACK (RAM)

	Pos	Val (D)	Addr

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)
R06	0
R07	0
R08	0
R09	0
R10	0
R11	0
R12	0
R13	0
R14	0
R15	0
R16	0
R17	0
R18	0
R19	0
R20	0
R21	0
R22	0
R23	0
R24	0
R25	0
R26	0
R27	0
R28	0
R29	0
R30	0
R31	0

Program | Instructions

New Program

Program Name

Pages: 1

Base Address

ADD

Programs

SAVE...

Program List: ALL

LOAD...

Base Address: -1

COPY TO CLIPBOARD

Program Control | CPU View | CPU Help

SWITCH TO ADVANCED CPU VIEW...

Registers | Program Stack

Reg Value

CHANGE

RESET ALL

Show Reg Access Status

Select Register Set Size: 32

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]

CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T

Cache - Pipeline | Execution Unit

Pipeline
- Single pipeline
- Dual pipeline
- Select pipeline: 0
- SHOW PIPELINE...

Cache
- Select cache type: Data
- SHOW CACHE...

SPECIAL CPU REGISTERS

PC	0	SR	0
SP	8096	BR	0

SR Status Flag: OV Z N

CPU Mode: User / Kernel

IR

MAR: 0

MDR

PROGRAM LIST

Name	Base	Start	Type

- LOAD COMPILED CODE IN MEMORY
- SHOW PROGRAM DATA MEMORY...
- REMOVE PROGRAM
- REMOVE ALL PROGRAMS
- CREATE PROGRAM INSTANCE
- DELETE PROGRAM INSTANCE

PROGRAM STACK (RAM)

	Pos	Val (D)	Addr

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)	C	Val (D)
R06	0		
R07	0		
R08	0		
R09	0		
R10	0		
R11	0		
R12	0		
R13	0		
R14	0		
R15	0		
R16	0		
R17	0		
R18	0		
R19	0		
R20	0		
R21	0		
R22	0		
R23	0		
R24	0		
R25	0		
R26	0		
R27	0		
R28	0		
R29	0		
R30	0		
R31	0		

Program | Instructions | Optimize - Assemble

New Program
- Program Name
- Pages: 1
- Base Address
- ADD

Programs
- SAVE...
- Program List: ALL
- LOAD...
- Base Address: -1
- COPY TO CLIPBOARD

Program Control | CPU View | CPU Help

- SWITCH TO SIMPLE CPU VIEW...

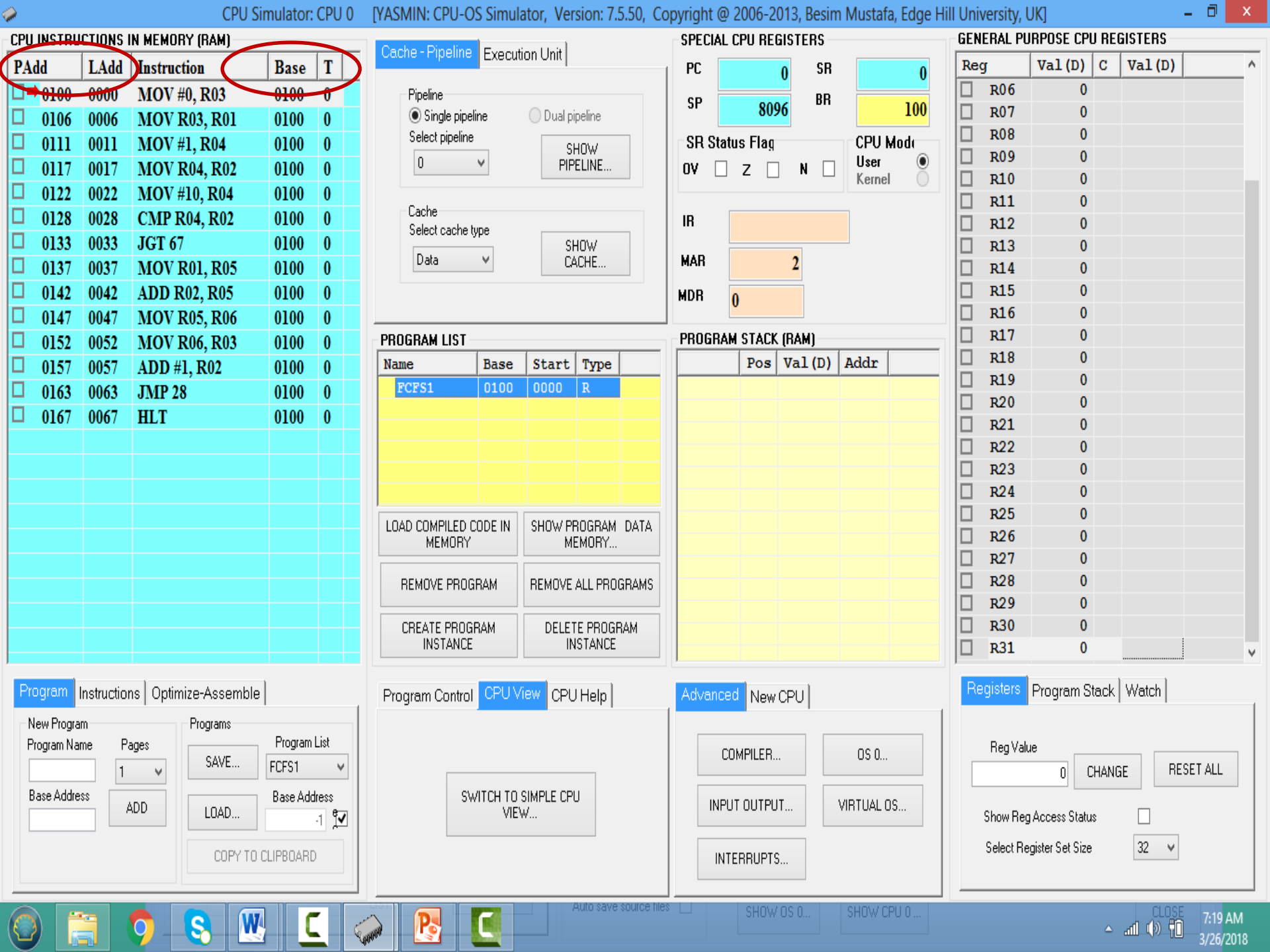
Advanced | New CPU

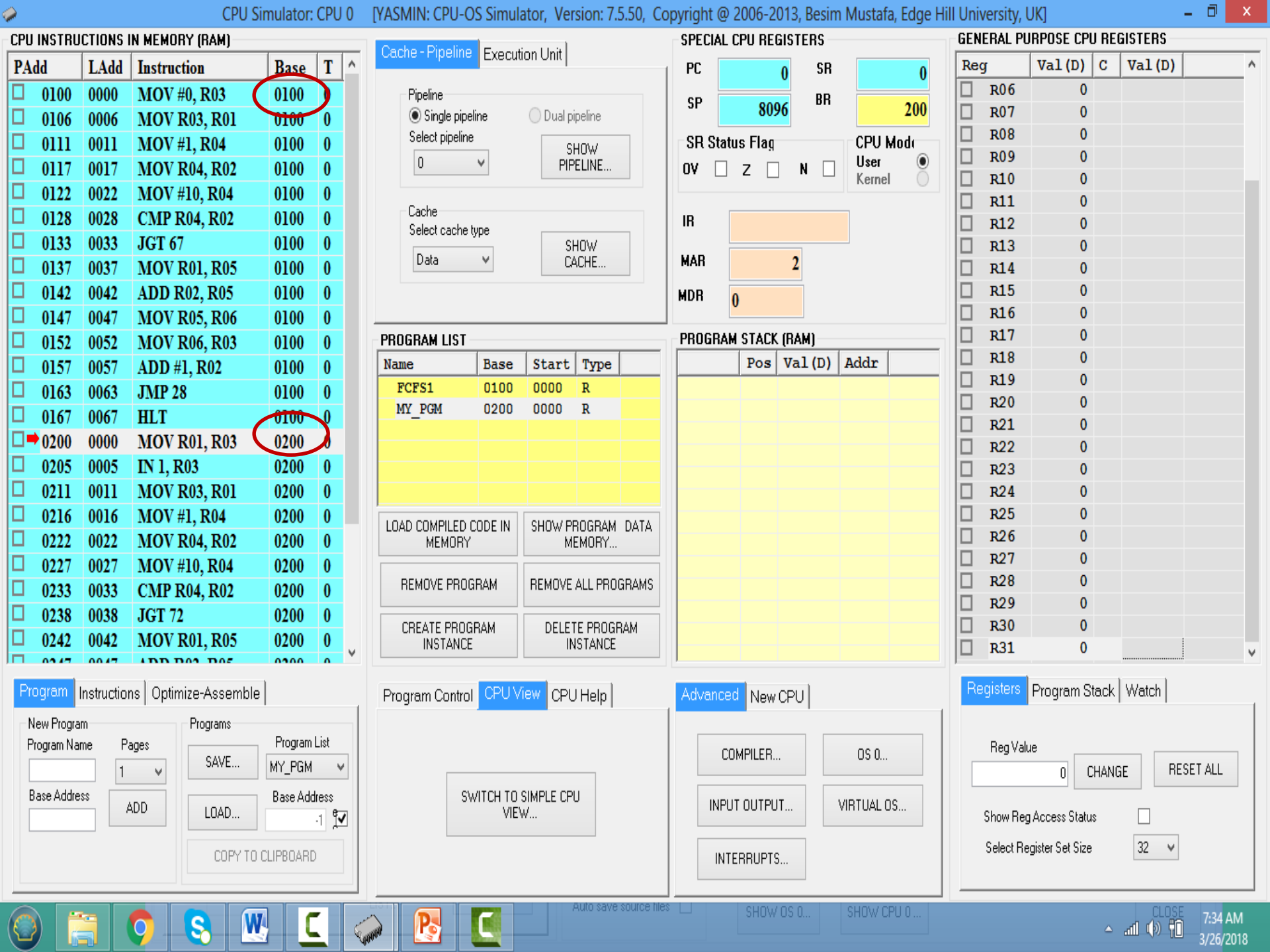
- COMPILER...
- OS 0...
- INPUT OUTPUT...
- VIRTUAL OS...
- INTERRUPTS...

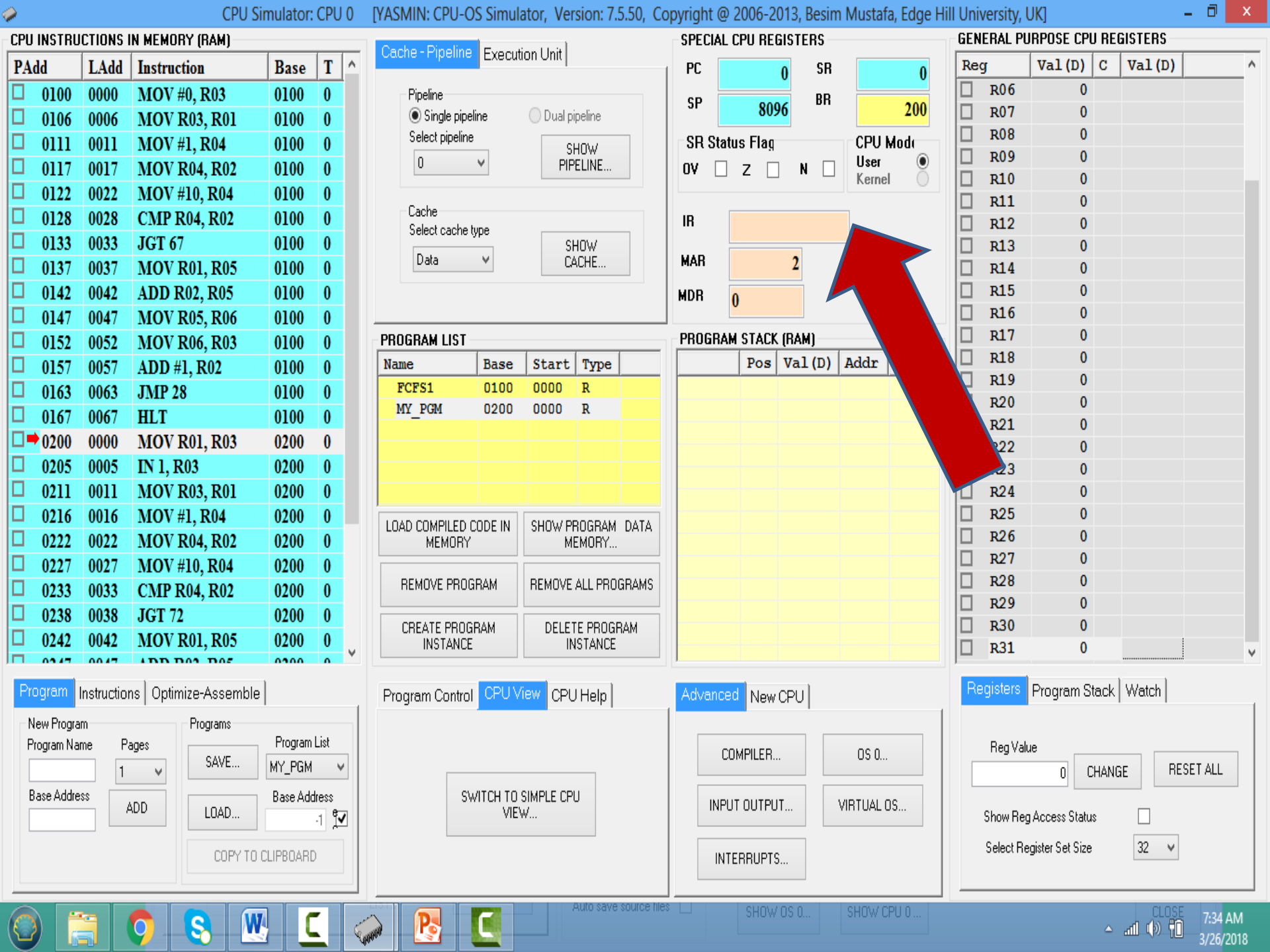
Registers | Program Stack | Watch

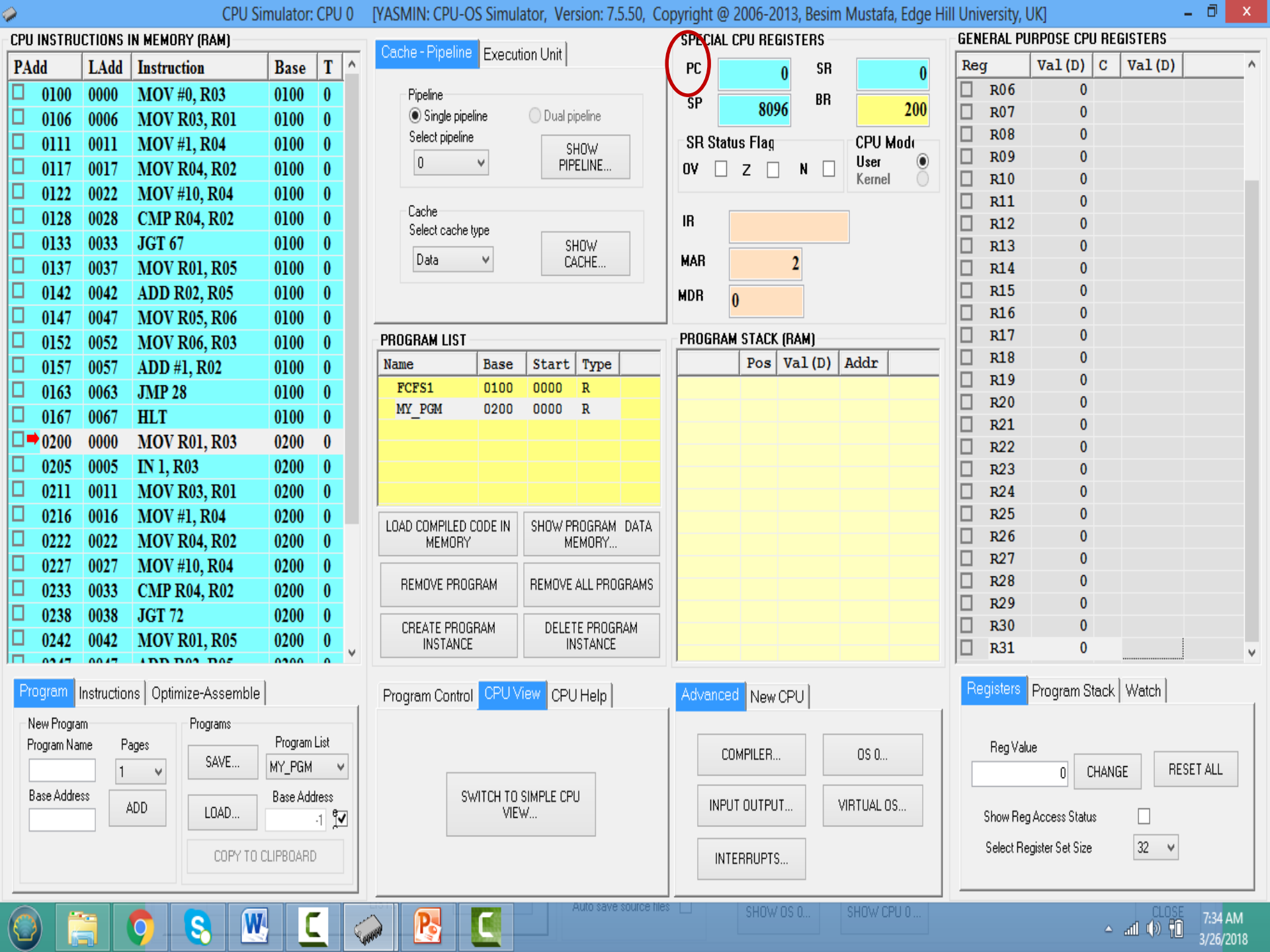
- Reg Value
- CHANGE
- RESET ALL
- Show Reg Access Status
- Select Register Set Size: 32

6:44 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 100
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
FCFS1
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:19 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]

CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T
0100	0000	MOV #0, R03	0100	0
0106	0006	MOV R03, R01	0100	0
0111	0011	MOV #1, R04	0100	0
0117	0017	MOV R04, R02	0100	0
0122	0022	MOV #10, R04	0100	0
0128	0028	CMP R04, R02	0100	0
0133	0033	JGT 67	0100	0
0137	0037	MOV R01, R05	0100	0
0142	0042	ADD R02, R05	0100	0
0147	0047	MOV R05, R06	0100	0
0152	0052	MOV R06, R03	0100	0
0157	0057	ADD #1, R02	0100	0
0163	0063	JMP 28	0100	0
0167	0067	HLT	0100	0
0200	0000	MOV R01, R03	0200	0
0205	0005	IN 1, R03	0200	0
0211	0011	MOV R03, R01	0200	0
0216	0016	MOV #1, R04	0200	0
0222	0022	MOV R04, R02	0200	0
0227	0027	MOV #10, R04	0200	0
0233	0033	CMP R04, R02	0200	0
0238	0038	JGT 72	0200	0
0242	0042	MOV R01, R05	0200	0

Cache - Pipeline | Execution Unit

Pipeline
Single pipeline | Dual pipeline
Select pipeline: 0
SHOW PIPELINE...

Cache
Select cache type: Data
SHOW CACHE...

SPECIAL CPU REGISTERS

PC: 0
SR: 0
SP: 8096
BR: 200

SR Status Flag
OV Z N

CPU Mode
User
Kernel

IR:
MAR: 2
MDR: 0

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)	C	Val (D)
R06	0		
R07	0		
R08	0		
R09	0		
R10	0		
R11	0		
R12	0		
R13	0		
R14	0		
R15	0		
R16	0		
R17	0		
R18	0		
R19	0		
R20	0		
R21	0		
R22	0		
R23	0		
R24	0		
R25	0		
R26	0		
R27	0		
R28	0		
R29	0		
R30	0		
R31	0		

PROGRAM LIST

Name	Base	Start	Type
FCFS1	0100	0000	R
MY_PGM	0200	0000	R

LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE

PROGRAM STACK (RAM)

	Pos	Val (D)	Addr

Program | Instructions | Optimize-Assemble

New Program
Program Name
Pages: 1
Base Address
ADD

Programs
SAVE...
Program List: MY_PGM
LOAD...
Base Address: -1
COPY TO CLIPBOARD

Program Control | CPU View | CPU Help

SWITCH TO SIMPLE CPU VIEW...

Advanced | New CPU

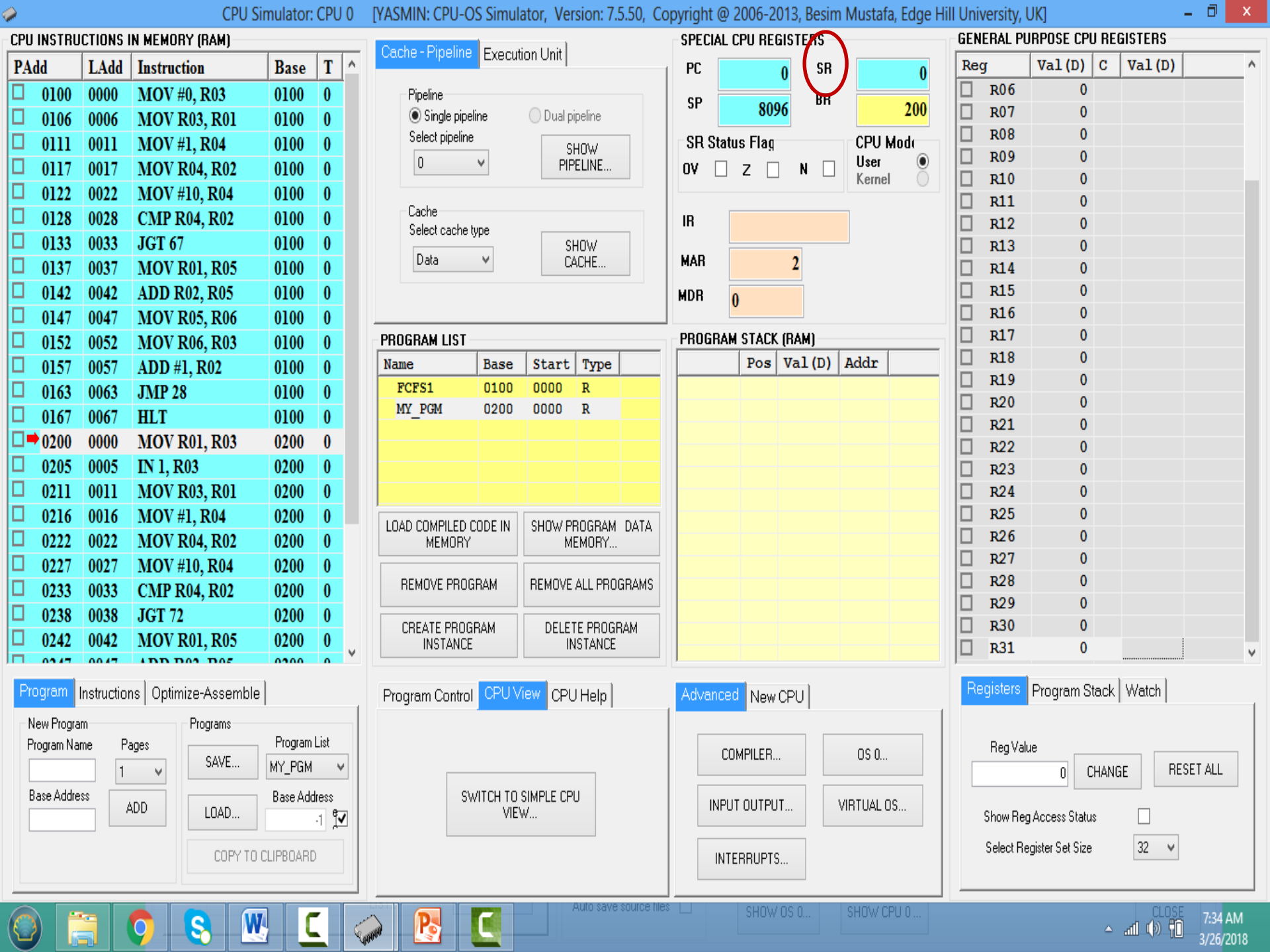
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...

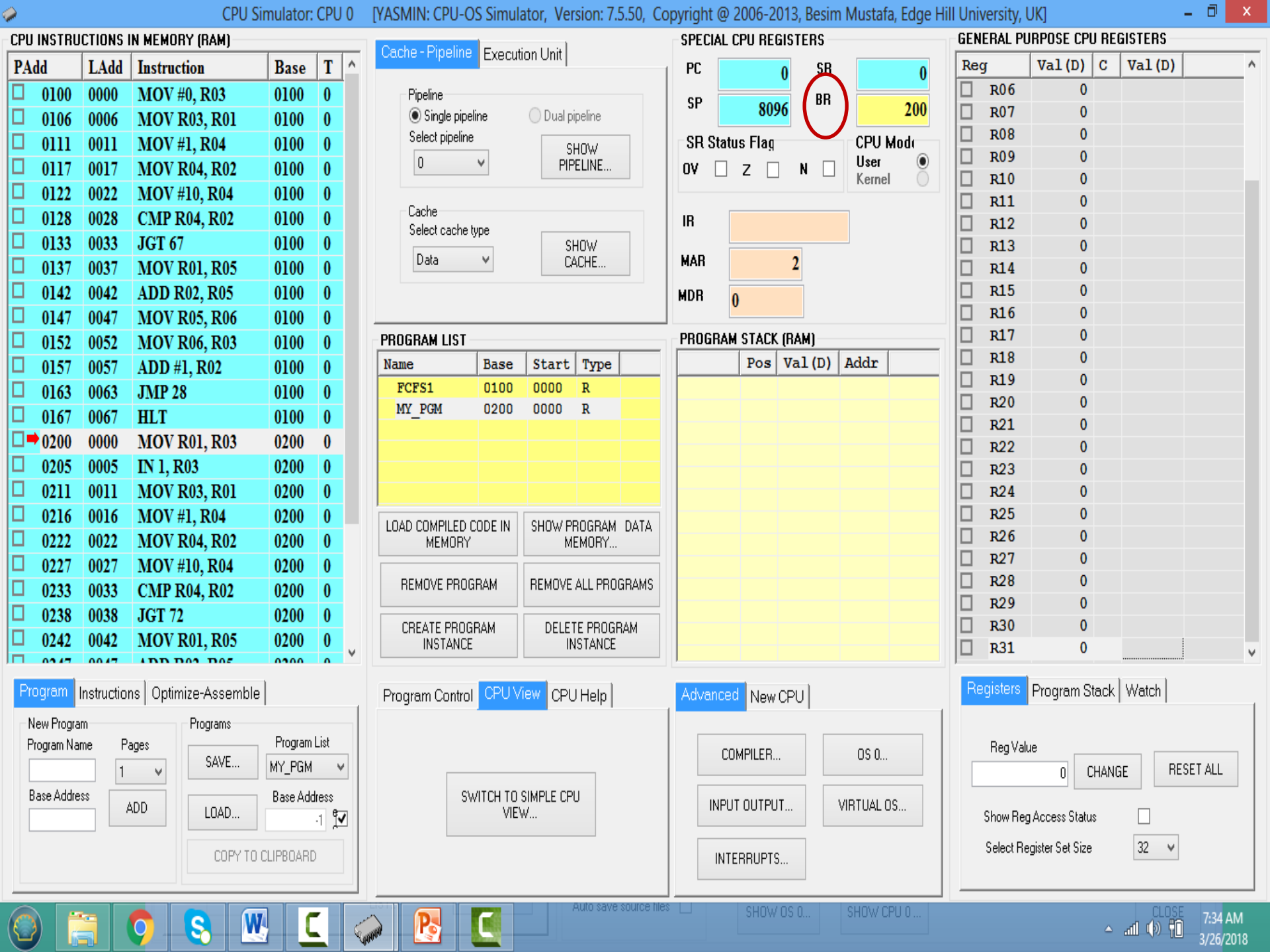
Registers | Program Stack | Watch

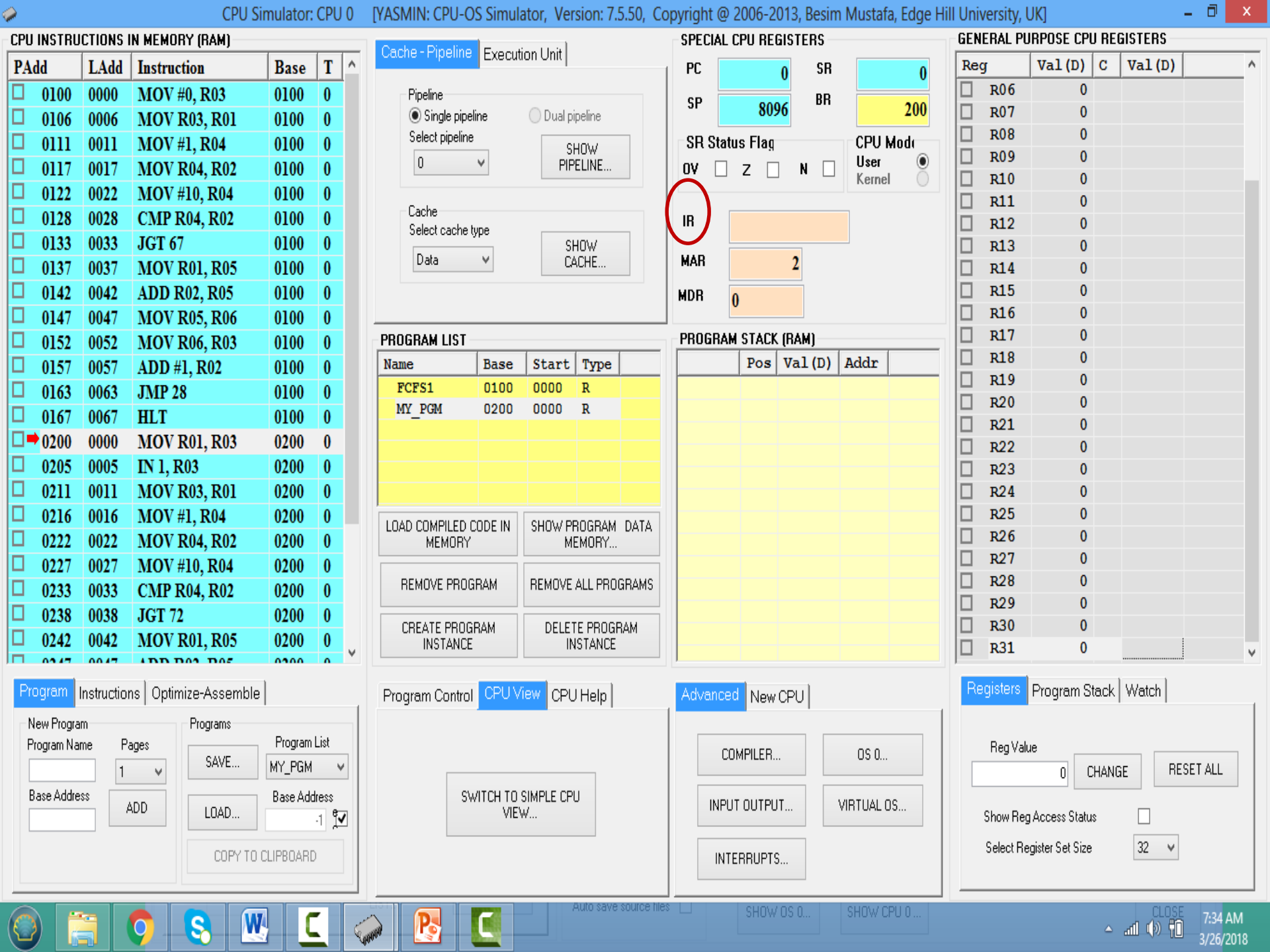
Reg Value: 0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size: 32

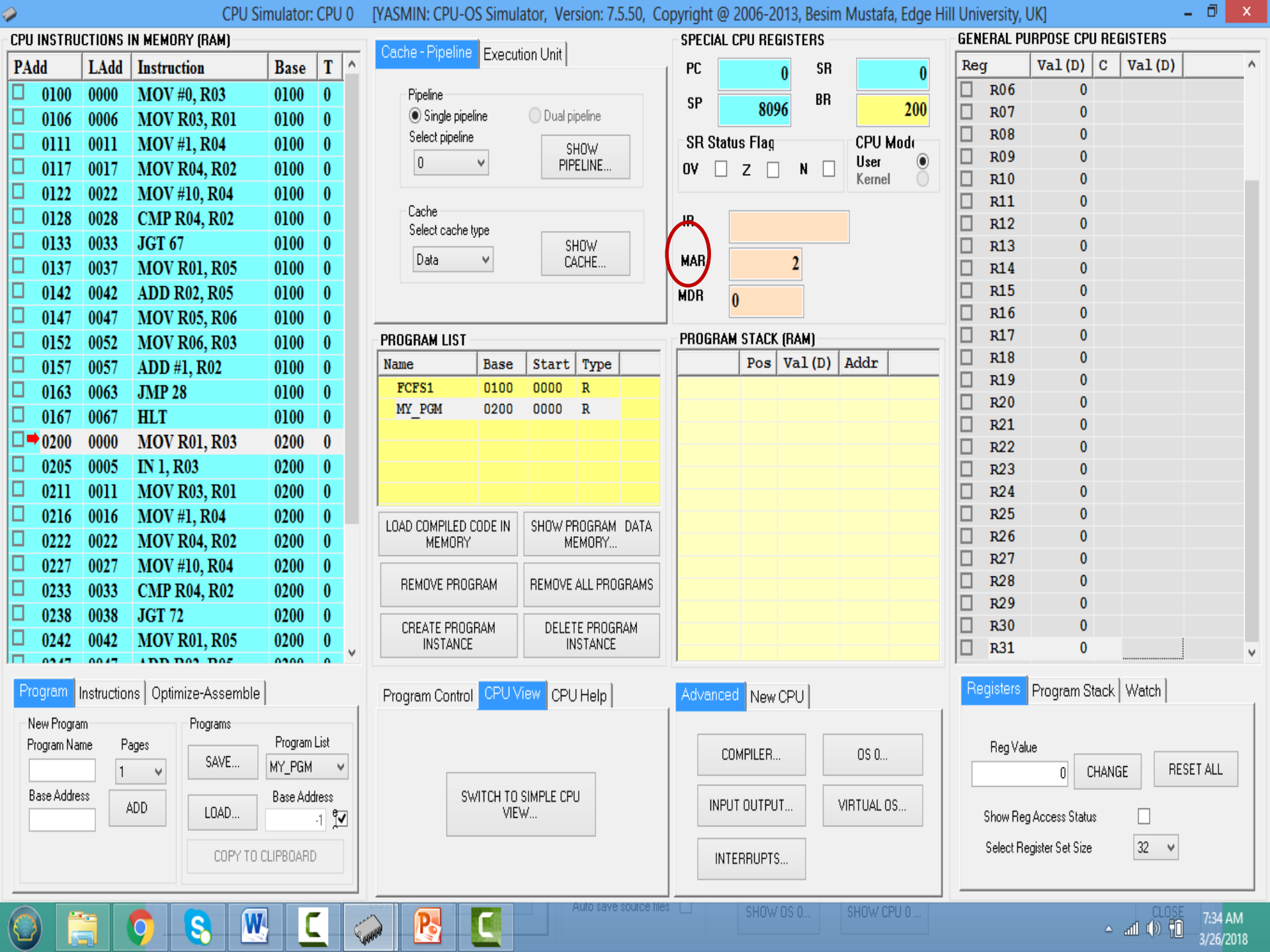
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE

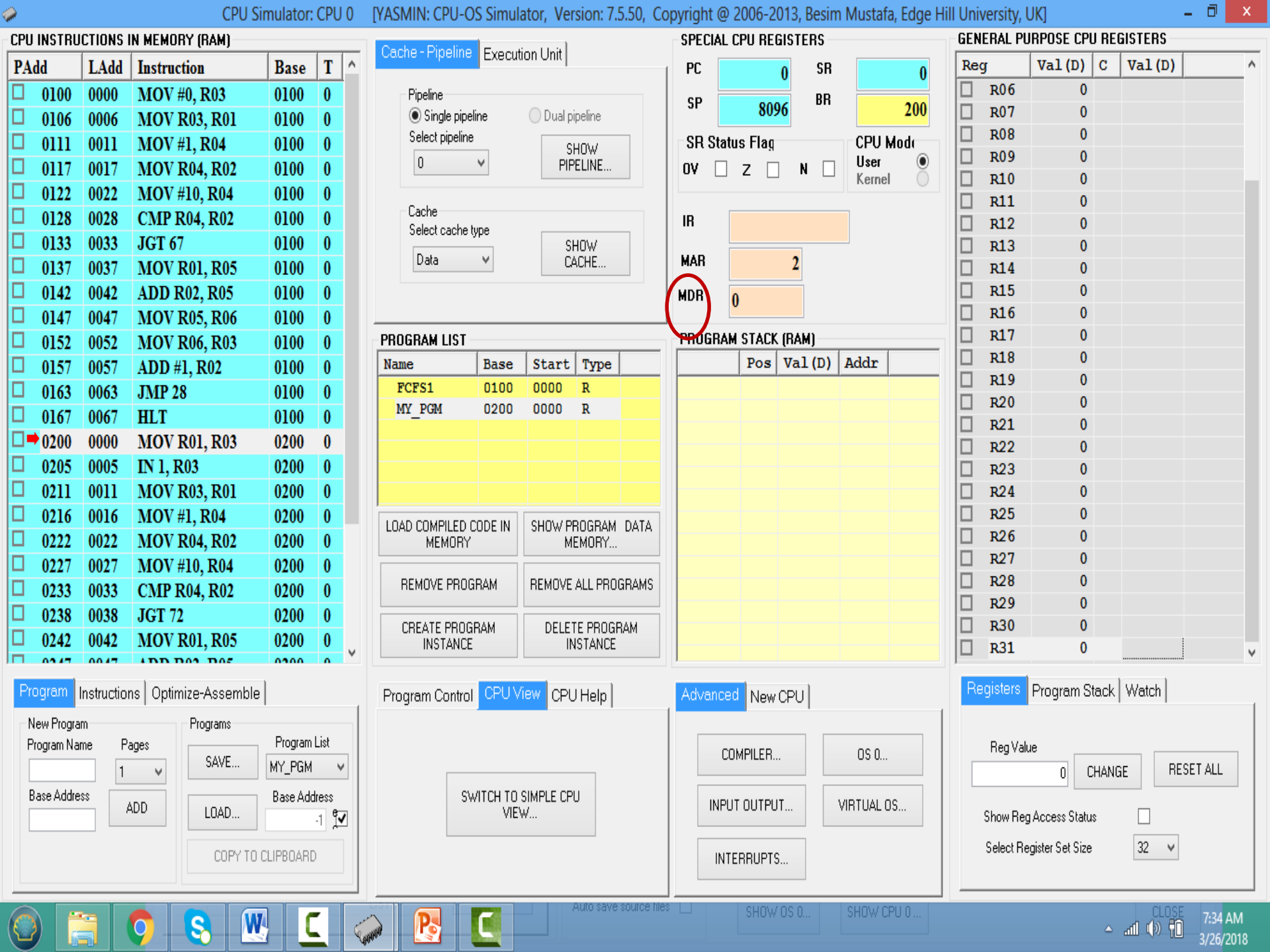
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]

CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T
0100	0000	MOV #0, R03	0100	0
0106	0006	MOV R03, R01	0100	0
0111	0011	MOV #1, R04	0100	0
0117	0017	MOV R04, R02	0100	0
0122	0022	MOV #10, R04	0100	0
0128	0028	CMP R04, R02	0100	0
0133	0033	JGT 67	0100	0
0137	0037	MOV R01, R05	0100	0
0142	0042	ADD R02, R05	0100	0
0147	0047	MOV R05, R06	0100	0
0152	0052	MOV R06, R03	0100	0
0157	0057	ADD #1, R02	0100	0
0163	0063	JMP 28	0100	0
0167	0067	HLT	0100	0
0200	0000	MOV R01, R03	0200	0
0205	0005	IN 1, R03	0200	0
0211	0011	MOV R03, R01	0200	0
0216	0016	MOV #1, R04	0200	0
0222	0022	MOV R04, R02	0200	0
0227	0027	MOV #10, R04	0200	0
0233	0033	CMP R04, R02	0200	0
0238	0038	JGT 72	0200	0
0242	0042	MOV R01, R05	0200	0

Cache - Pipeline | Execution Unit

Pipeline
Single pipeline | Dual pipeline
Select pipeline: 0
SHOW PIPELINE...

Cache
Select cache type: Data
SHOW CACHE...

SPECIAL CPU REGISTERS

PC: 0
SR: 0
SP: 8096
BR: 200

SR Status Flag
OV Z N

CPU Mode
User
Kernel

IR:
MAR: 2
MDR: 0

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)	C	Val (D)
R06	0		
R07	0		
R08	0		
R09	0		
R10	0		
R11	0		
R12	0		
R13	0		
R14	0		
R15	0		
R16	0		
R17	0		
R18	0		
R19	0		
R20	0		
R21	0		
R22	0		
R23	0		
R24	0		
R25	0		
R26	0		
R27	0		
R28	0		
R29	0		
R30	0		
R31	0		

PROGRAM LIST

Name	Base	Start	Type
FCFS1	0100	0000	R
MY_PGM	0200	0000	R

LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE

PROGRAM STACK (RAM)

	Pos	Val (D)	Addr

Program | Instructions | Optimize-Assemble

New Program
Program Name
Pages: 1
Base Address
ADD

Programs
SAVE...
Program List: MY_PGM
LOAD...
Base Address: -1
COPY TO CLIPBOARD

Program Control | CPU View | CPU Help

SWITCH TO SIMPLE CPU VIEW...

Advanced | New CPU

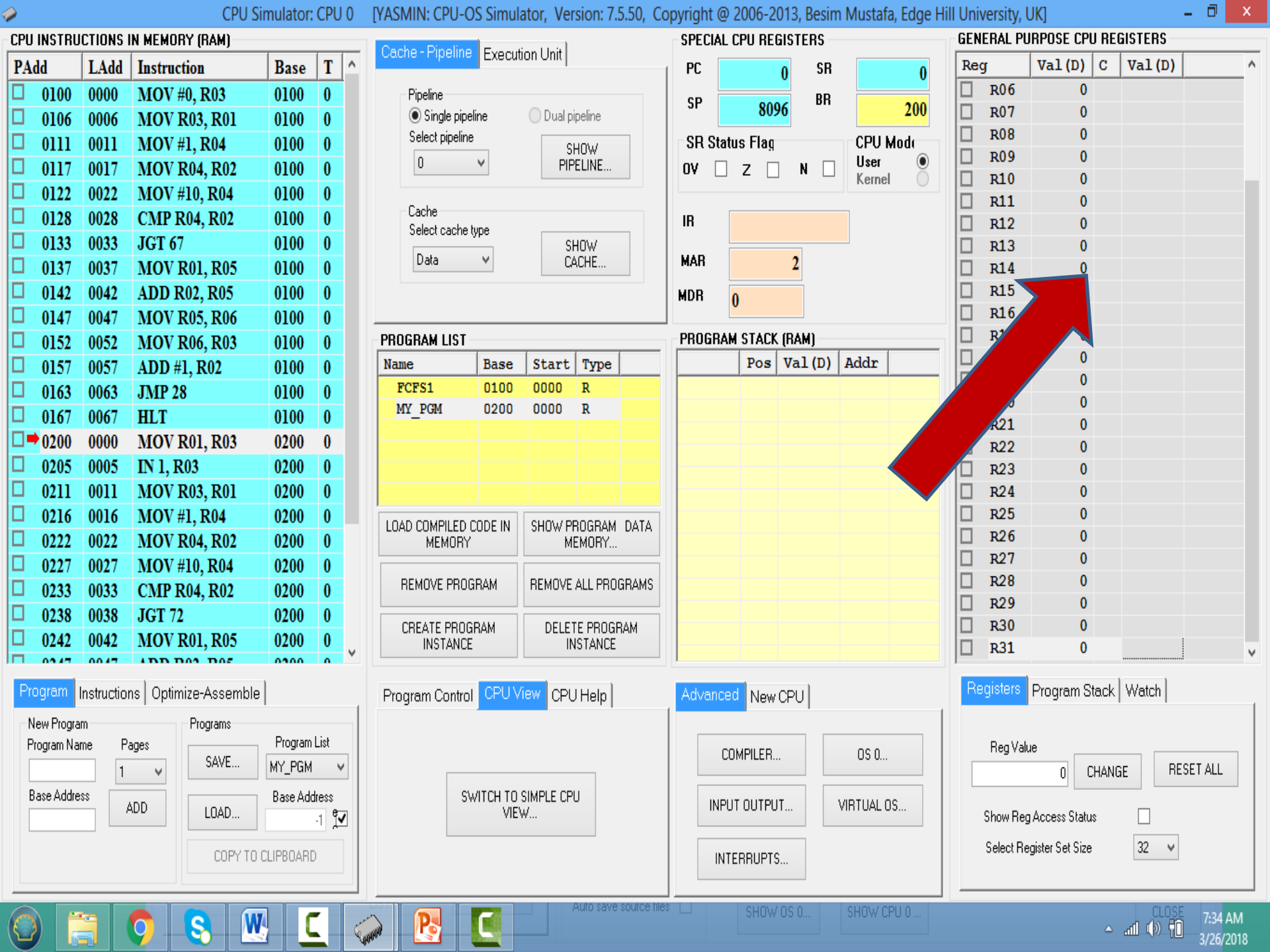
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...

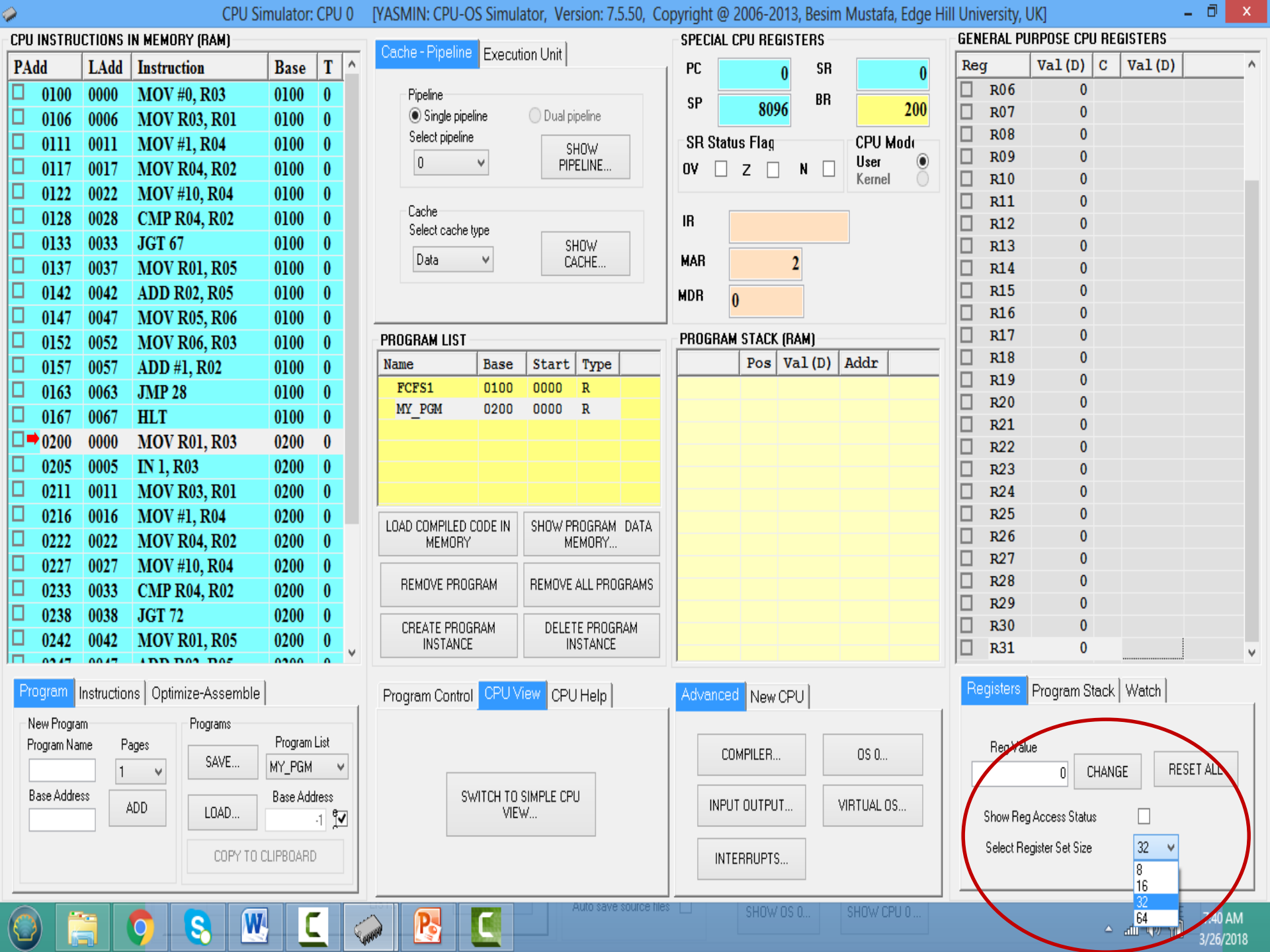
Registers | Program Stack | Watch

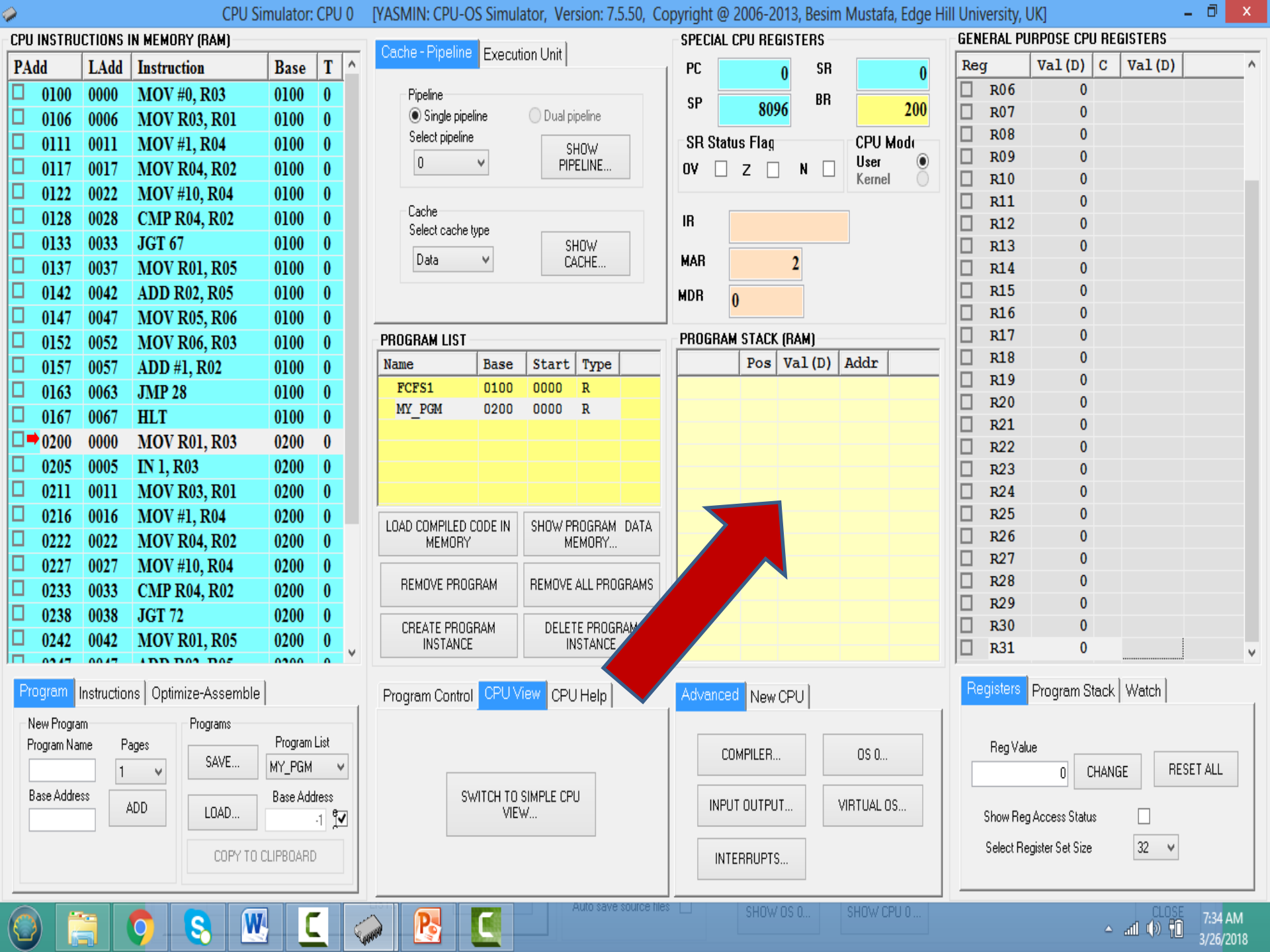
Reg Value: 0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size: 32

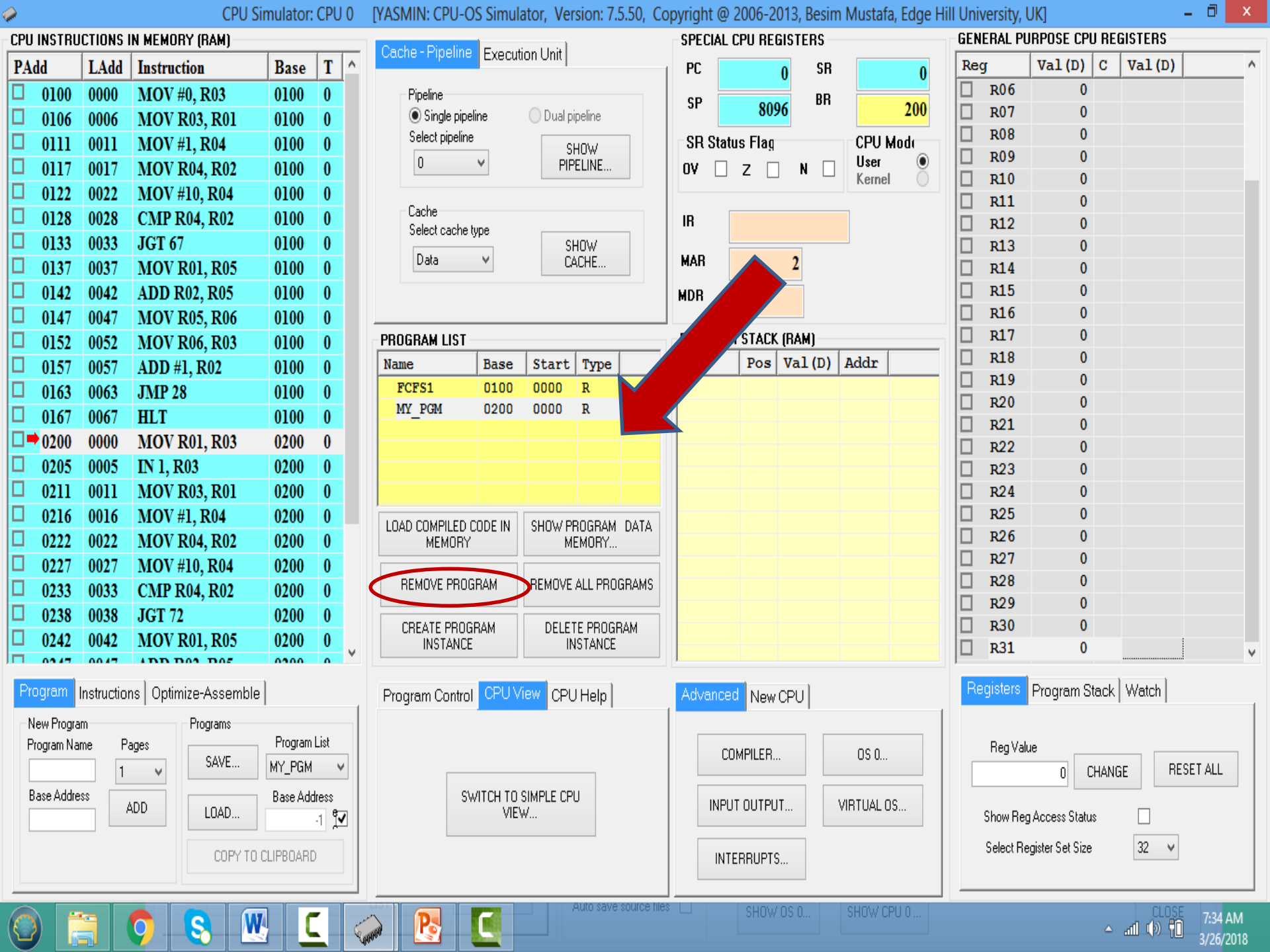
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE

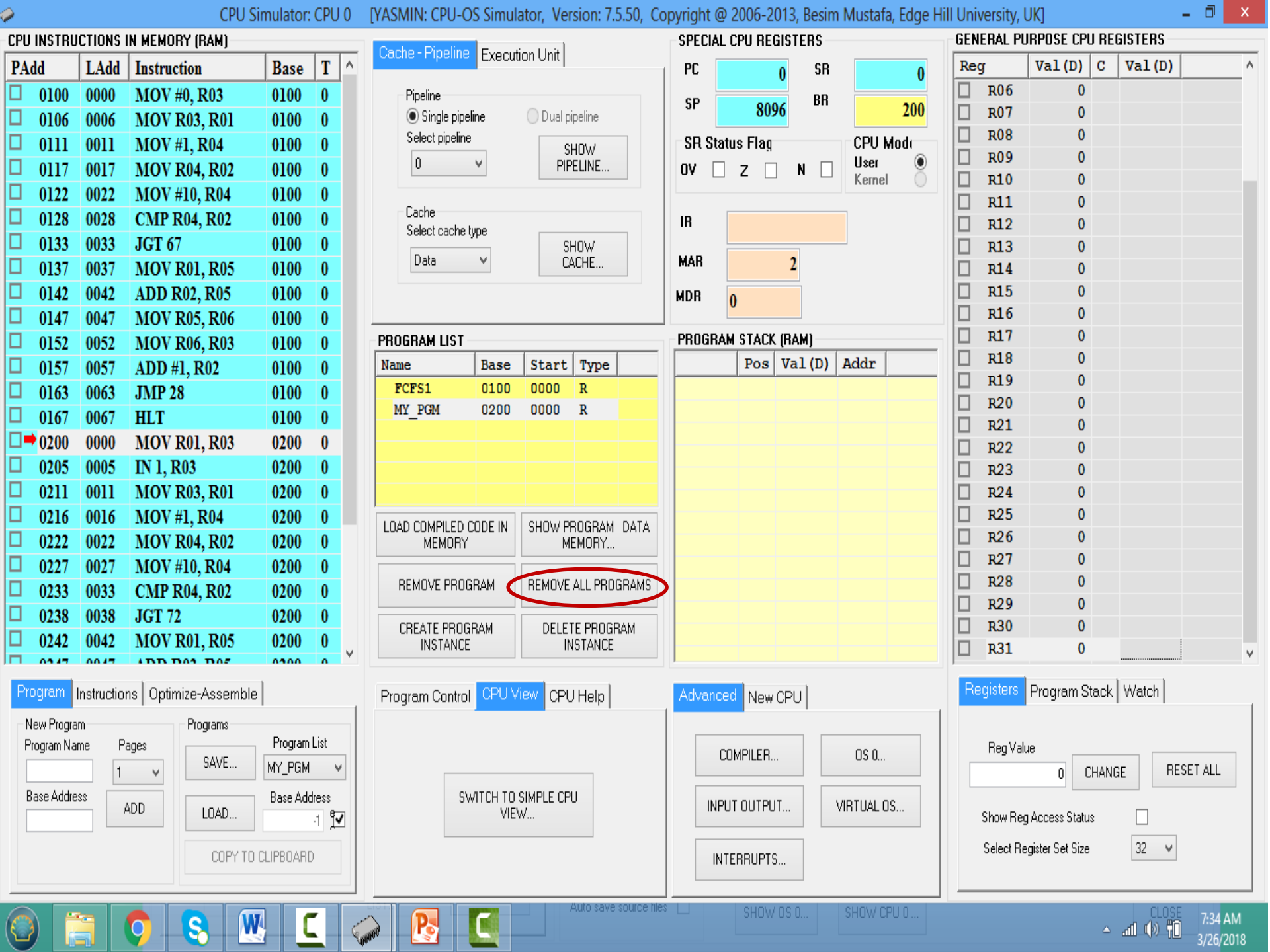
7:34 AM
3/26/2018

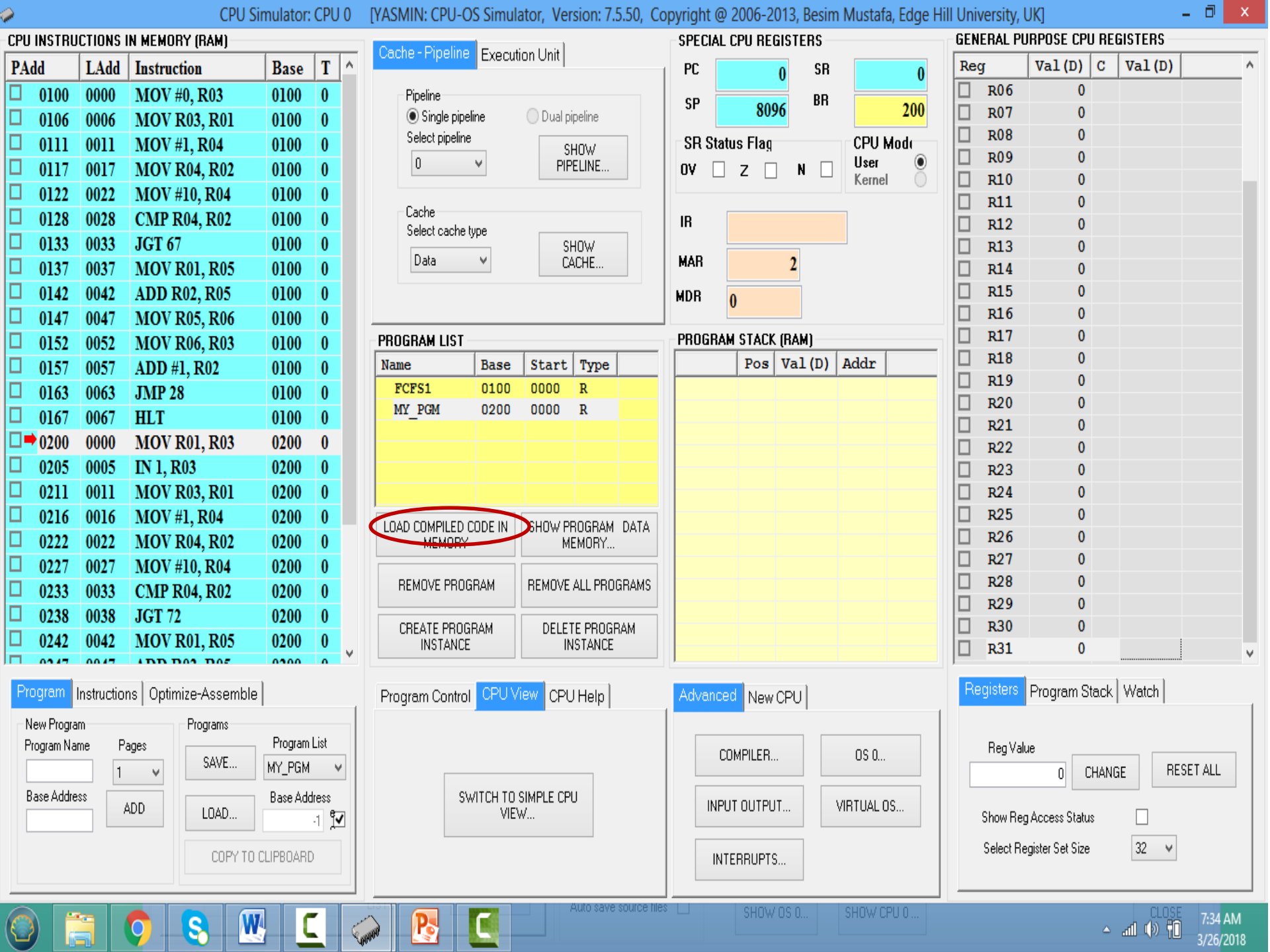

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018

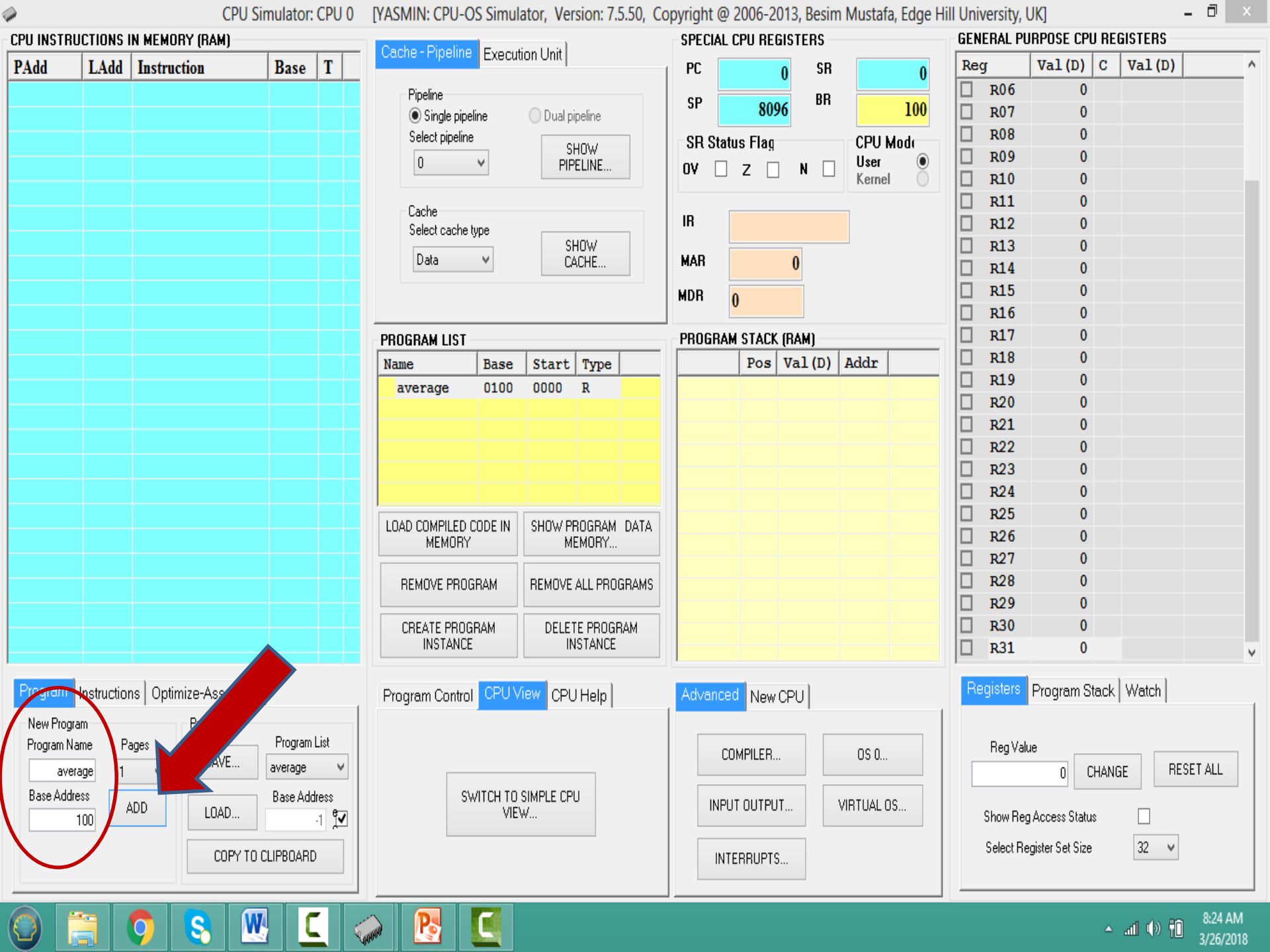

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
8
16
32
64
Auto save source files
SHOW OS 0...
SHOW CPU 0...
7:40 AM
3/26/2018

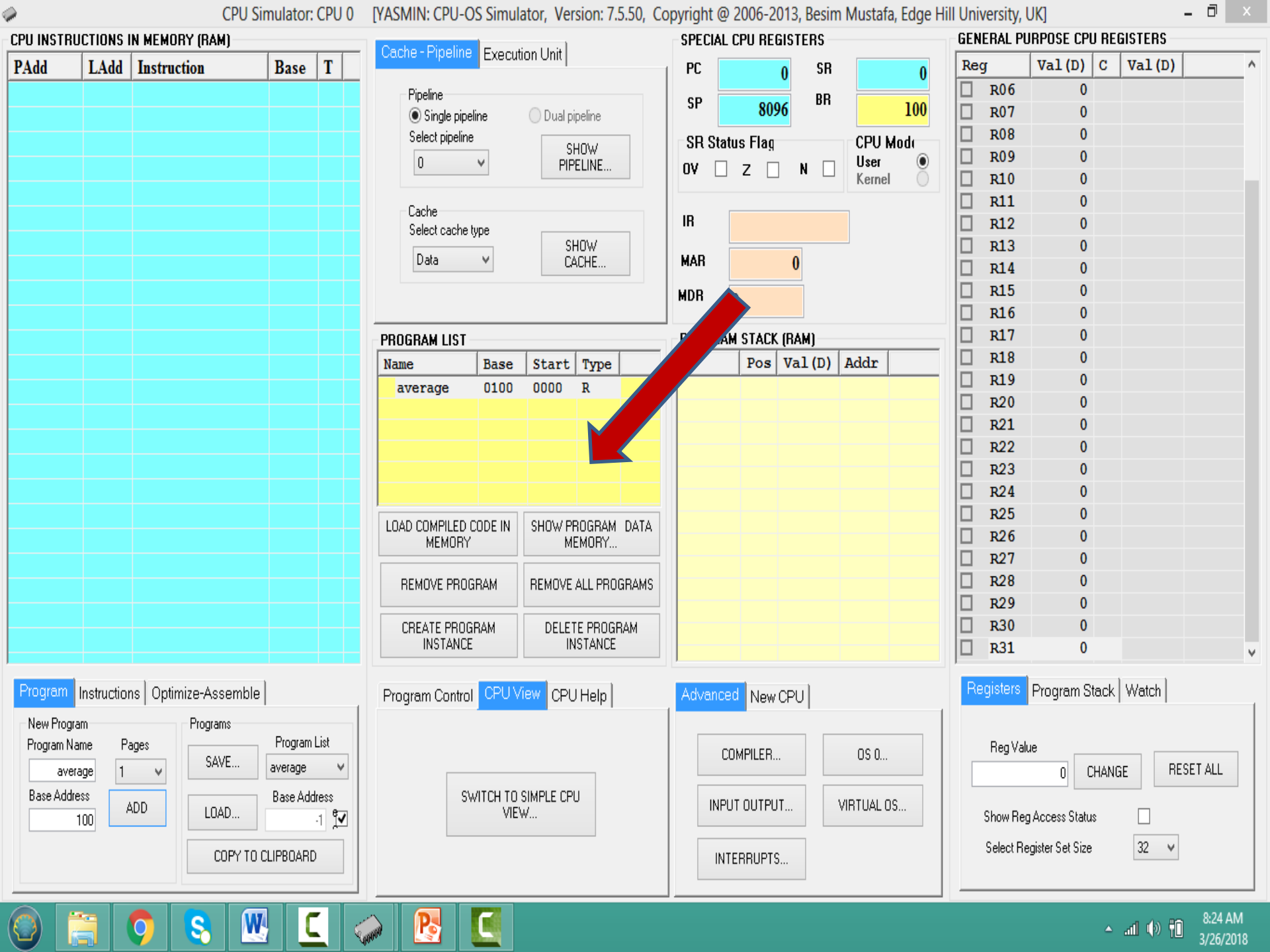

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018

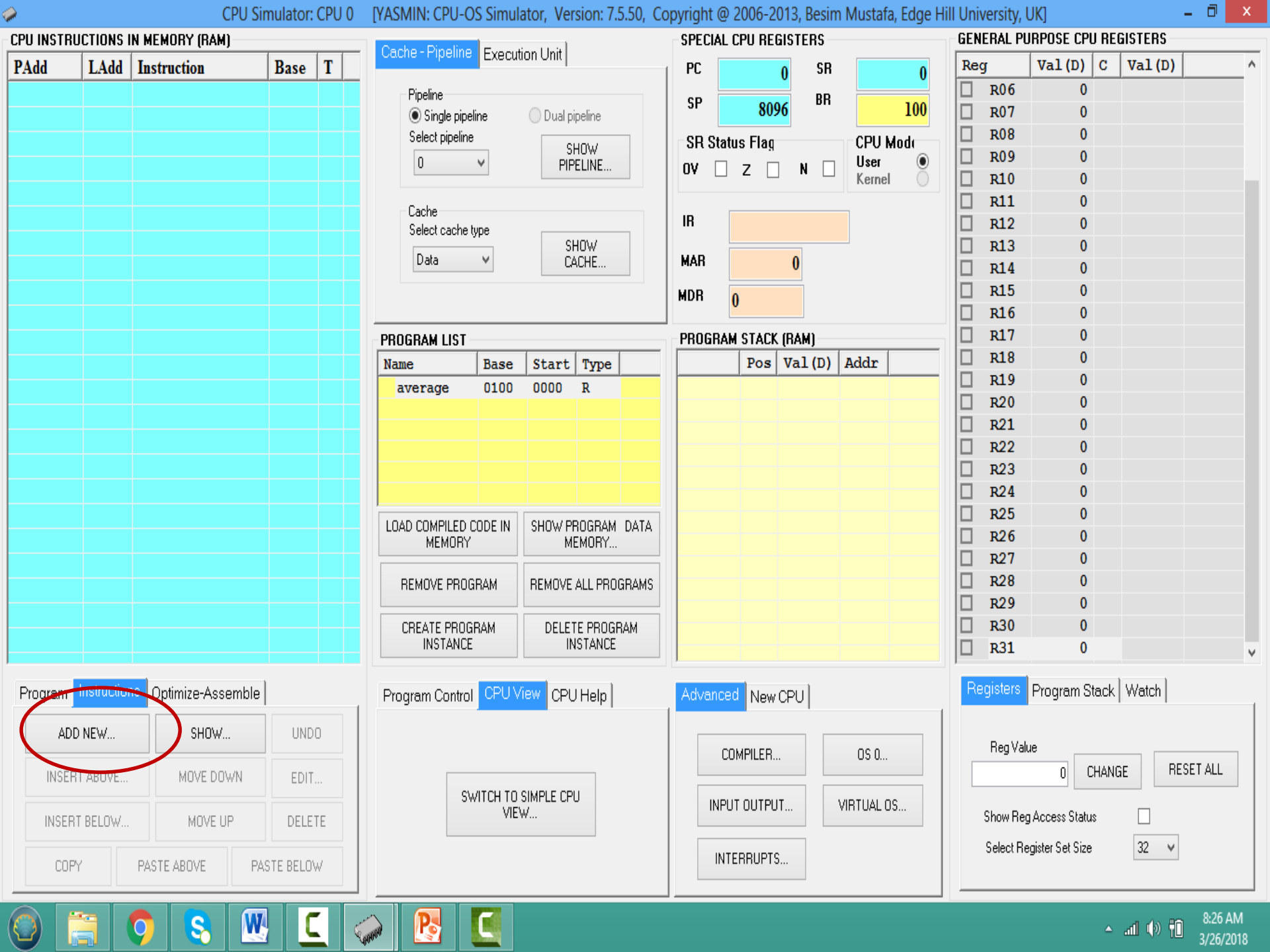

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
STACK (RAM)
Pos Val (D) Addr
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018

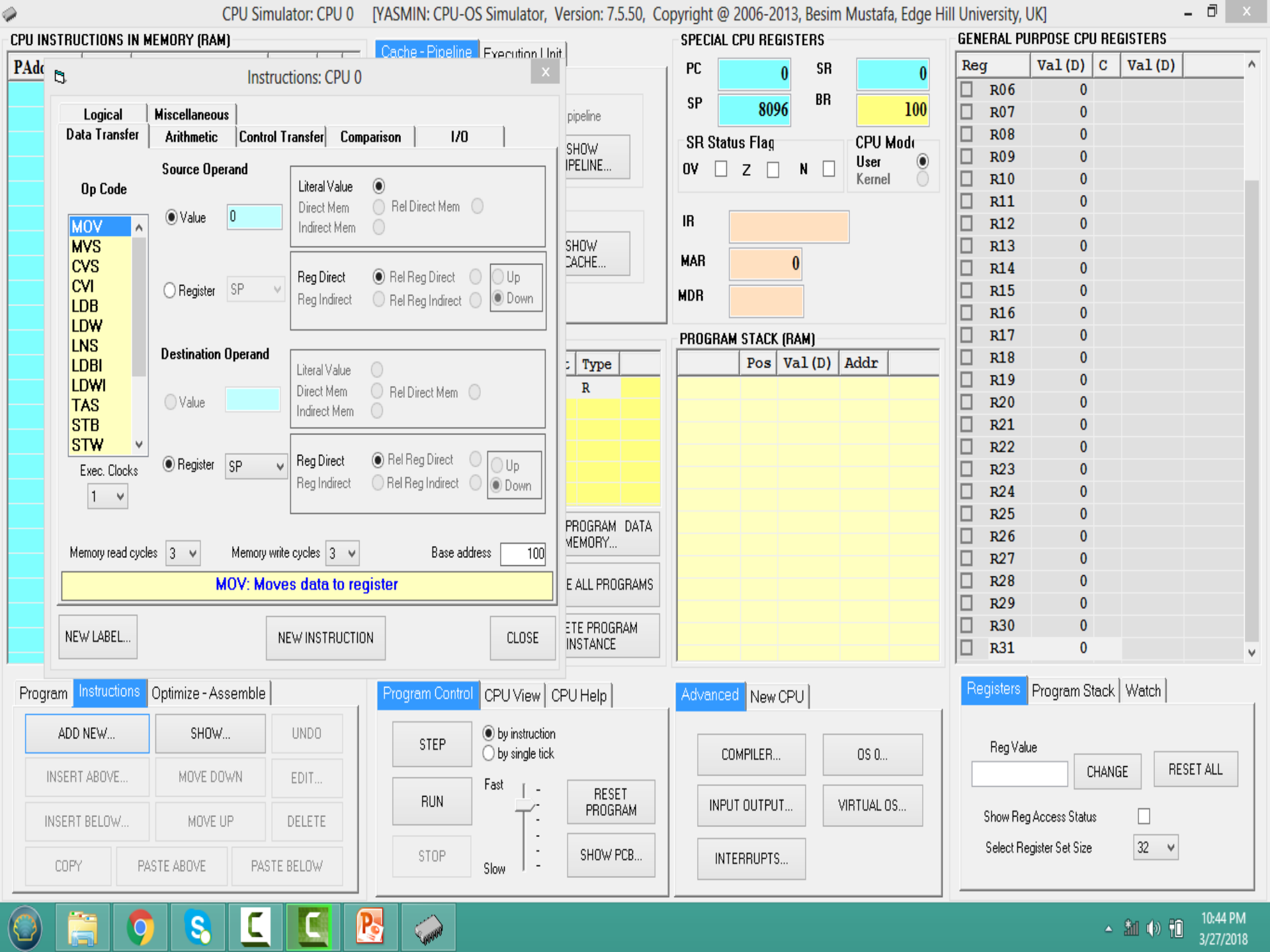

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018

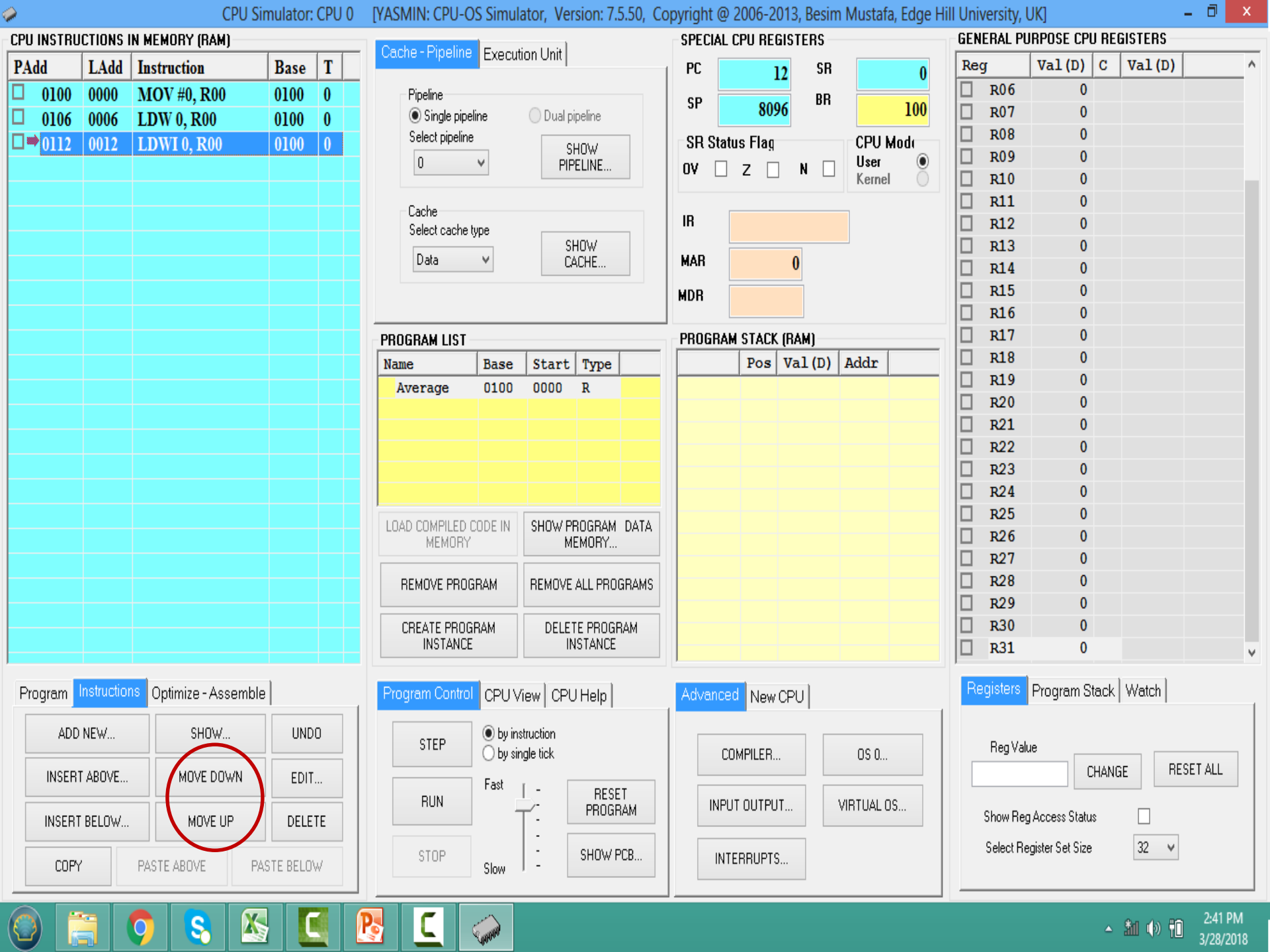

CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R03 0100 0
0106 0006 MOV R03, R01 0100 0
0111 0011 MOV #1, R04 0100 0
0117 0017 MOV R04, R02 0100 0
0122 0022 MOV #10, R04 0100 0
0128 0028 CMP R04, R02 0100 0
0133 0033 JGT 67 0100 0
0137 0037 MOV R01, R05 0100 0
0142 0042 ADD R02, R05 0100 0
0147 0047 MOV R05, R06 0100 0
0152 0052 MOV R06, R03 0100 0
0157 0057 ADD #1, R02 0100 0
0163 0063 JMP 28 0100 0
0167 0067 HLT 0100 0
0200 0000 MOV R01, R03 0200 0
0205 0005 IN 1, R03 0200 0
0211 0011 MOV R03, R01 0200 0
0216 0016 MOV #1, R04 0200 0
0222 0022 MOV R04, R02 0200 0
0227 0027 MOV #10, R04 0200 0
0233 0033 CMP R04, R02 0200 0
0238 0038 JGT 72 0200 0
0242 0042 MOV R01, R05 0200 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 0
SR 0
SP 8096
BR 200
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 2
MDR 0
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
FCFS1 0100 0000 R
MY_PGM 0200 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize-Assemble
New Program
Program Name
Pages
1
Base Address
ADD
Programs
SAVE...
Program List
MY_PGM
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
Auto save source files
SHOW OS 0...
SHOW CPU 0...
CLOSE
7:34 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd
LAdd
Instruction
Base
T
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC
0
SR
0
SP
8096
BR
100
SR Status Flag
OV
Z
N
CPU Mode
User
Kernel
IR
MAR
0
MDR
0
GENERAL PURPOSE CPU REGISTERS
Reg
Val (D)
C
Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name
Base
Start
Type
average
0100
0000
R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos
Val (D)
Addr
Program
Instructions
Optimize-Ass
New Program
Program Name
average
Pages
1
Base Address
100
ADD
AVE...
LOAD...
Program List
average
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
8:24 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd
LAdd
Instruction
Base
T
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC
0
SR
0
SP
8096
BR
100
SR Status Flag
OV
Z
N
CPU Mode
User
Kernel
IR
MAR
0
MDR
GENERAL PURPOSE CPU REGISTERS
Reg
Val (D)
C
Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name
Base
Start
Type
average
0100
0000
R
AM STACK (RAM)
Pos
Val (D)
Addr
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
Program
Instructions
Optimize-Assemble
New Program
Program Name
average
Pages
1
Base Address
100
ADD
Programs
SAVE...
Program List
average
LOAD...
Base Address
-1
COPY TO CLIPBOARD
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
8:24 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd
LAdd
Instruction
Base
T
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
0
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC
0
SR
0
SP
8096
BR
100
SR Status Flag
OV
Z
N
CPU Mode
User
Kernel
IR
MAR
0
MDR
0
GENERAL PURPOSE CPU REGISTERS
Reg
Val (D)
C
Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name
Base
Start
Type
average
0100
0000
R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos
Val (D)
Addr
Program
Instructions
Optimize-Assemble
ADD NEW...
SHOW...
UNDO
INSERT ABOVE...
MOVE DOWN
EDIT...
INSERT BELOW...
MOVE UP
DELETE
COPY
PASTE ABOVE
PASTE BELOW
Program Control
CPU View
CPU Help
SWITCH TO SIMPLE CPU VIEW...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
0
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
8:26 AM
3/26/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
Cache - Pipeline
Execution Unit
Instructions: CPU 0
Logical
Miscellaneous
Data Transfer
Arithmetic
Control Transfer
Comparison
I/O
Op Code
MOV
MVS
CVS
CVI
LDB
LDW
LNS
LDBI
LDWI
TAS
STB
STW
Exec. Clocks
Source Operand
Value
Register
SP
Literal Value
Direct Mem
Indirect Mem
Rel Direct Mem
Reg Direct
Reg Indirect
Rel Reg Direct
Rel Reg Indirect
Up
Down
Destination Operand
Memory read cycles
Memory write cycles
Base address
MOV: Moves data to register
NEW LABEL...
NEW INSTRUCTION
CLOSE
SPECIAL CPU REGISTERS
PC
SP
SR
BR
SR Status Flag
OV
Z
N
CPU Mode
User
Kernel
IR
MAR
MDR
PROGRAM STACK (RAM)
Pos
Val (D)
Addr
Type
GENERAL PURPOSE CPU REGISTERS
Reg
Val (D)
C
Program
Instructions
Optimize - Assemble
ADD NEW...
SHOW...
UNDO
INSERT ABOVE...
MOVE DOWN
EDIT...
INSERT BELOW...
MOVE UP
DELETE
COPY
PASTE ABOVE
PASTE BELOW
Program Control
CPU View
CPU Help
STEP
by instruction
by single tick
RUN
STOP
Fast
Slow
RESET PROGRAM
SHOW PCB...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
10:44 PM
3/27/2018


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
0100 0000 MOV #0, R00 0100 0
0106 0006 LDW 0, R00 0100 0
0112 0012 LDWI 0, R00 0100 0
Cache - Pipeline
Execution Unit
Pipeline
Single pipeline
Dual pipeline
Select pipeline
SHOW PIPELINE...
Cache
Select cache type
Data
SHOW CACHE...
SPECIAL CPU REGISTERS
PC 12
SR 0
SP 8096
BR 100
SR Status Flag
OV Z N
CPU Mode
User
Kernel
IR
MAR 0
MDR
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
R06 0
R07 0
R08 0
R09 0
R10 0
R11 0
R12 0
R13 0
R14 0
R15 0
R16 0
R17 0
R18 0
R19 0
R20 0
R21 0
R22 0
R23 0
R24 0
R25 0
R26 0
R27 0
R28 0
R29 0
R30 0
R31 0
PROGRAM LIST
Name Base Start Type
Average 0100 0000 R
LOAD COMPILED CODE IN MEMORY
SHOW PROGRAM DATA MEMORY...
REMOVE PROGRAM
REMOVE ALL PROGRAMS
CREATE PROGRAM INSTANCE
DELETE PROGRAM INSTANCE
PROGRAM STACK (RAM)
Pos Val (D) Addr
Program
Instructions
Optimize - Assemble
ADD NEW...
SHOW...
UNDO
INSERT ABOVE...
MOVE DOWN
EDIT...
INSERT BELOW...
MOVE UP
DELETE
COPY
PASTE ABOVE
PASTE BELOW
Program Control
CPU View
CPU Help
STEP
by instruction
by single tick
RUN
Fast
Slow
RESET PROGRAM
STOP
SHOW PCB...
Advanced
New CPU
COMPILER...
OS 0...
INPUT OUTPUT...
VIRTUAL OS...
INTERRUPTS...
Registers
Program Stack
Watch
Reg Value
CHANGE
RESET ALL
Show Reg Access Status
Select Register Set Size
32
2:41 PM
3/28/2018

Instruction set architecture

7 groups:

- Data Transfer instructions
- Arithmetic instructions
- Logical instructions
- Control transfer instructions
- Comparison instructions
- I/O instructions
- Miscellaneous instructions

Data Transfer Instructions

- **MOV SRC DST**
  - Move SRC to DST
  - Move immediate Data to Register
  - Move Register to Register
- Example:
  - MOV #2, R01 ➔ Move number 2 in to register R01
  - MOV R01, R02 ➔ Move contents of register R01 to R02

Example1.sas

Data Transfer Instructions

- **STB SRC MEM**
  - Store a byte in to memory
  - SRC can be a register or immediate value
  - Address of the memory can specified directly or indirectly
- Example: Direct Memory Addressing
  - STB R02, 0001, ➔ Stores a byte in register R02 to memory whose address is 0001
- Example: Indirect Memory / Indirect Register Addressing
  - STB R02, @R08 ➔ Stores a byte in register R02 to memory whose address is in Register R08
- Example: Immediate Source
  - STB #02, 0002

Example2.sas

Data Transfer Instructions

- **STW SRC MEM**
  - Store a WORD (2 Bytes) in to memory
  - SRC can be a register or immediate value
  - Address of the memory can specified directly or indirectly
- Example: Direct Addressing
  - STW R02, 0000, ➔ Stores a byte in register R02 to memory whose address is 0000
- Example: Indirect Addressing
  - STW R02, @R08 ➔ Stores a byte in register R02 to memory whose address is in Register R08
- Example: Immediate Source
  - STW #1023, 0002

Example3_1.sas

Example3_2.sas

Data Transfer Instructions

- **LDB MEM REG**
  - Load a byte from memory to register
  - Address of the memory can specified directly indirectly
- Example: Direct Addressing
  - LDB 0001, R01 ➔ Loads a byte from memory address 0001 into register R01
- Example: Indirect Addressing
  - LDB @R02, R01 ➔ Loads a byte from memory location whose address is in Register R02 into register R01

Example4.sas

Data Transfer Instructions

- **LDW MEM REG**
  - Load a byte from memory to register
  - Address of the memory can specified directly indirectly
- Example: Direct Addressing
  - LDW 0001, R01 ➔ Loads a word from memory address 0001 into register R01
- Example: Indirect Addressing
  - LDW @R02, R01 ➔ Loads a word from memory location whose address is in Register R02 into register R01

Example5.sas

Example5_1.sas

Data Transfer Instructions

- **PUSH**
  - Push data to top of hardware stack (TOS); push register to TOS
  - Examples:
    - **PSH #6** ;pushes number 6 on top of the stack
    - **PSH R03** ;pushes the contents of register R03 on top of the stack

Data Transfer Instructions

- **POP**
  - Pop data from top of hardware stack to register
  - Example:
    - **POP R05** ;pops contents of top of stack into register R05

Example6.sas

Data Transfer Instructions

- **MVS SRC DST**
  - Moves a string from SRC to DST
- Addressing modes for SRC and DST
  - Direct
  - Indirect
  - Register Indirect

Example7.sas

Exercise…

Edit Example7.sas with the following:

1) MVS 0, 40
2) MVS 1, 40
3) MVS @16, 40

Data Transfer Instructions

- **SWAP REG1 REG2**
  - Swaps the contents of two registers.
  - Only register addressing mode
- Example:
  - SWAP R01, R02

Example8.sas

Data Transfer Instructions

- **LDBI**
  - Loads Byte from memory into register, increments indirect source address
  - Example:
    - LDBI @R00, R01
- **LDWI**
  - Loads word from memory into register, increments indirect source address
  - Example:
    - LDWI @R00, R01

Example9.sas

Data Transfer Instructions

- **STBI**
  - Stores Byte to memory, increments indirect destination address
  - Example:
    - STBI R01, @R00
- **STWI**
  - Stores word to memory, increments indirect destination address
  - Example:
    - STWI R01, @R00

Arithmetic Instructions

- **ADD SRC DST**
  - Adds values in registers or adds immediate value to register
  - DST ← DST + SRC
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - ADD R01, R02
    - ADD #1, RO3

Arithmetic Instructions

- **SUB SRC DST**
  - subtracts values in registers or subtracts immediate value from a register
  - DST ← DST - SRC
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - SUB R01, R02
    - SUB #1, RO3

Arithmetic Instructions

- **SUBU SRC DST**
  - subtracts values in registers or subtracts immediate value from a register with **unsigned result**
  - DST ← DST - SRC
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - SUBU R01, R02
    - SUBU #1, RO3

Arithmetic Instructions

- **MUL SRC DST**
  - Multiply values in SRC and DST
  - DST ← DST * SRC
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - MUL R01, R02
    - MUL #1, RO3

Arithmetic Instructions

- **DIV SRC DST**
  - Divides values in SRC and DST
  - DST ← DST / SRC
  - Remainder is lost and quotient is saved in DST
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - DIV R01, R02
    - DIV #1, RO3

Arithmetic Instructions

- **INC SRC**
  - Increments value in SRC by 1
  - SRC ← SRC + 1
- **DEC SRC**
  - Decrements value in SRC by 1
  - SRC ← SRC - 1
- SRC uses Register Addressing mode
- Example:
  - INC R01

Logical Instructions

AND

Inputs		Output
A	B	F
0	0	0
1	0	0
0	1	0
1	1	1

- **AND SRC DST**
  - Logical AND of values in SRC and DST
  - DST ← DST & SRC
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - AND R01, R02
    - AND #1, RO3

Logical Instructions

OR

Inputs		Output
A	B	F
0	0	0
1	0	1
0	1	1
1	1	1

- **OR SRC DST**
  - Logical OR of values in SRC and DST
  - DST ← DST | SRC
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - OR R01, R02
    - OR #1, RO3

Logical Instructions

NOT

Input	Output
I	F
0	1
1	0

- **NOT SRC DST**
  - Logical NOT of value in SRC
  - DST ← !SRC
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - NOT R01 R02
    - NOT #5, R02

Logical Instructions

- **SHL CNT DST**
  - Shift to left value in DST
  - DST ← Shift Left DST (CNT times)
  - CNT can be Immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - SHL 2, R00
    - SHL R01, R02

Example_11.sas

Logical Instructions

- **SHR CNT DST**
  - Shift to Right value in DST
  - DST ← Shift Right DST (CNT times)
  - CNT can be Immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - SHR 2, R00
    - SHR R01, R02

Example_11.sas

Comparison Instructions

- **CMP SRC DST**
  - Compares SRC and DST contents
  - If
    - DST = SRC ➔ Zero flag is set
    - DST < SRC ➔ Negative flag is set
    - DST > SRC ➔ Zero and Negative flags are reset
  - SRC can be immediate or Register Addressing mode but uses only register addressing mode for DST
  - Example:
    - CMP R00, R01
    - CMP #5, R01

Example_12.sas

Comparison Instructions

- **CPS SRC DST**
  - Compares SRC and DST strings
  - If
    - DST = SRC ➔ Zero flag is set
    - DST < SRC ➔ Negative flag is set
    - DST > SRC ➔ Zero and Negative flags are reset
  - SRC and DST can be Direct Memory or register indirect addressing mode
  - Example:
    - CPS 0, 16
    - CPS @R01, @R02
    - CPS @R01, 8

Example_13.sas

Control Transfer Instructions

innovate achieve lead

Instruction	Description	Example
JMP TARGET	Jumps Unconditionally	JMP 100
JLT TARGET	Jumps if less than	JLT 100 jumps to address location 100 if the previous comparison instruction result has set the **N** status flag.
JGT TARGET	Jumps if greater than	JGT 100 jumps to address location 100 if the previous comparison instruction result has not set the **N** and **Z** status flag.
JEQ TARGET	Jumps if equal	JEQ 100 jumps to address location 100 if the previous comparison instruction result has set **Z** status flag.
JNE TARGET	Jumps if not equal	JNE 100 jumps to address location 100 if the previous comparison instruction result indicates DST is less than SRC (i.e. the **N** status flag is set).

Control Transfer Instructions

Instruction	Description	Example
JGE TARGET	Jumps if greater than or equal	JGE 100
JLE TARGET	Jumps if less than or equal	JLE 100
JNZ TARGET	Jumps if not zero	JNZ 100
JZR TARGET	Jumps if zero	JZR 100
LOOP TARGET, REG	Loops if REG value is greater than 0	LOOP 100, R00

Control Transfer Instructions

- HLT
  - Halts the simulator

Subroutine

- CAL <SUB_ADDRESS>
  - Calls a subroutine
- RET
  - Returns from subroutine
  - The last instruction in a subroutine must always be the following instruction
  - Execution of RET instruction retrieves the context and will jump to the instruction
- MSF
  - Mark Stack Frame

Example

Write a program to add two numbers using a subroutine.

```
START :    MOV #05, R00
           MOV #06, R01
           MSF
           CAL <ADD_SUB>
           STB R03, 8
           HLT
ADD_SUB:   ADD R00, R01
           MOV R01, R03
           RET
```

Example_14.sas

I/O Instructions

- **IN PORT_NO DESTINATION**
  - Gets input into register or memory from I/O Device
  - PORT_NO for keyboard is 00
  - Destination can be register or memory
- **OUT SOURCE PORT_NO**
  - Puts output from register or memory to I/O device
  - PORT_NO for monitor is 01
  - Source can be register or memory

Miscellaneous Instructions

- NOP
  - No operation

Next Webinar…

innovate achieve lead

- Thursday, 29/11/2018 @ 7.30 PM
- Topics to be covered
  - Cache Memory
    - Direct Mapped
    - Associative Cache
    - Set Associative Cache
  - Replacement Policy
  - Write Policy
- Resource Person : Prof. Chandrashekar R K

Thank You

BITS Pilani

Pilani Campus