pkg default

CISCGUI
- simu : CISCSimulator

- initComponents() : void
- loadUserInstruction() : boolean
- inputUserData() : boolean
- buttonIPLActionPerformed(evt : ActionEvent) : void
- buttonRunActionPerformed(evt : ActionEvent) : void
- buttonSaveActionPerformed(evt : ActionEvent) : void
- buttonSingleStepActionPerformed(evt : ActionEvent) : void
- buttonHaltActionPerformed() : void
- buttonLoadProgram1ActionPerformed() : void
- buttonLoadProgram2ActionPerformed() : void
+ uploadDisplay() : void

CISCSimulator
- controller : CPU
- memory : Memory
- panel : CISCGUI
- ioc : IOC
- state : StateType

+ initProcessor() : boolean
+ powerOff() : void
+ run() : boolean
+ singleStep() : boolean
+ setUserCode(arrInst : String[]) : void
+ inputUserText(text : String) : void
+ loadProgram(id : int, path : String) : boolean
+ loadRegistor(R0 : long, R1 : long, R2 : long, R3 : long, IX1 : long, IX2 : long, IX3 : long, IR : long, PC : long, CC : long, MAR : long, MBR : long, MFR : long) : void

IOC
+ appendIOBuffer(devID : int, i : char) : void
+ getIOBuffer(devID : int) : char
+ isIOBuffer(devID : int) : boolean
+ getStatus(devID : int) : boolean

ROM
+ getBinCode() : ArrayList<WORD>

CPU
- MAR : WORD
- MBR : WORD
- MFR : WORD
- IR : WORD
- PC : GBitSet
- GPR : GBitSet[4]
- IX : GBitSet[4]
- CC : GBitSet
- state : CPUState
- ih : InstructionHandler
- memory : Memory
- alu : ALU
- cache : Cache

+ init() : boolean
+ clock() : boolean
+ isCurrentInstruction() : boolean
- execute() : boolean
- isNextInstruction() : boolean
+ setBootCode() : boolean
+ setUserCode(arrAsmCode : String[]) : boolean
+ storeMemory(address : long, value : WORD) : boolean
+ loadMemory(address : long) : WORD
+ setTrap(code : int) : boolean
+ setFault(id : int) : boolean
+ increasePC() : boolean
+ getInputChar(devID : int) : char
+ setOutputChar(devID : int, out : char) : boolean

Memory
- memory : WORD[]
- DEFAULT_MEMORY : int
+ BOOT_MEMORY_START : int
+ USER_MEMORY_START : int
+ USER_PROGRAM_START : int

+ getUserMemoryLocation() : int
+ init() : boolean
+ setMemory(size : int) : boolean
+ load(address : int) : WORD
+ store(address : int, input : WORD, isSystem : boolean) : boolean
+ storeBootCode(arrCode : ArrayList<WORD>) : boolean
+ storeUserCode(arrCode : ArrayList<WORD>) : boolean

Cache
+ load(address : int) : WORD
+ store(address : int, input : WORD) : boolean
+ getCacheLine(tag : long) : CacheLine
+ addCacheLine(tag : long) : CacheLine

ALU
+ add(s1 : WORD, s2 : WORD) : WORD
+ sub(s1 : WORD, s2 : WORD) : WORD
+ mul(s1 : WORD, s2 : WORD) : WORD[]
+ div(s1 : WORD, s2 : WORD) : WORD[]
+ complement(s1 : WORD) : WORD
+ equal(s1 : WORD, s2 : WORD) : boolean
+ and(s1 : WORD, s2 : WORD) : WORD
+ or(s1 : WORD, s2 : WORD) : WORD
+ shift(s1 : WORD, count : int, isLeft : boolean, isArith : boolean) : WORD
+ rotate(s1 : WORD, count : int, isLeft : boolean, isArith : boolean) : WORD
+ setOverflow(input : boolean) : void
+ setUnderlow(input : boolean) : void
+ setDivzero(input : boolean) : void
+ setEqualOrNot(input : boolean) : void

InstructionHandler
- instSet : Hashtable<Integer,Instruction>
- textToCode : Hashtable<String,Integer>
- opcode : int
- reg : int
- ireg : int
- flag : int
- address : int
- IR : WORD
- lsInst : LSInstHandler
- alInst : ALInstHandler
- trInst : TransInstHandler
- ioInst : IOInstHandler

+ execute() : boolean
- parseIR(ir : WORD) : boolean
+ getAsmCode() : String
+ getBinCode(asmCode : String) : WORD
+ getEA() : int
+ getEAWithoutIX() : int

InstructionSet
+ getInstruction(int opcode : int) : Instruction

GBitSet
- length : int
- minValue : int
- maxValue : int

+ GBitSet(length : int) : void
+ setLong(number : long) : boolean
+ getLong() : long
+ copy(input : GBitSet) : boolean
+ subSet(from : int, to : int) : GBitSet
+ getString() : String

LSInstHandler
+ execute() : boolean
- executeLDA() : boolean
- executeSTR() : boolean
- executeLDX() : boolean
- executeSTX() : boolean
- executeLDR() : boolean

ALInstHandler
+ execute() : boolean
- executeAMR() : boolean
+ executeSMR() : boolean
+ executeAIR() : boolean
+ executeSIR() : boolean
+ executeMLT() : boolean
+ executeDVD() : boolean
+ executeTRR() : boolean
+ executeAND() : boolean
+ executeORR() : boolean
+ executeNOT() : boolean
+ executeSRC() : boolean
+ executeRRC() : boolean

TransInstHandler
+ execute() : boolean
- executeJZ() : boolean
+ executeJNE() : boolean
+ executeJGE() : boolean
+ executeJMA() : boolean
+ executeSOB() : boolean
+ executeJSR() : boolean
+ executeJCC() : boolean
+ executeRFS() : boolean

IOInstHandler
+ execute() : boolean
- executeIN() : boolean
+ executeOUT() : boolean
+ executeHLT() : boolean
+ executeCHK() : boolean
+ executeTRAP() : boolean

WORD
- SIZE : int

SignedWORD

sd CPU.clock

| CPU | InstructionHandler | Memory | Instruction | LSInstHandler | ALInstHandler | IOInstHandler | TransInstHandler |

alt

[state==LOAD_MAR]

1: result = isNextInstruction()

alt

[result==false]

2: [state=NO_INST]

[else]

3: [MAR=PC]

4: [state=LOAD_MBR]

[state==LOAD_MBR]

5: load(MAR)

result

6: [MBR=result]

7: [state=LOAD_IR]

[state==LOAD_IR]

8: [IR=MBR]

9: [state=EXECUTE]

[state==EXECUTE]

10: execute()

10.1: getType()

Instruction Type

alt

[type==Load/Store Instruction]

11: execute()

[type== Arithmetic Logical Instruction]

12: execute()

[type==IO or Halt Instruction]

13: execute()

[type==transfer Instruction]

14: execute()

sd IO

CPU	IOInstHandler	IOC	CISCGUI

1: execute()()

1.1: getInput()

1.1.1: getInput()

2: setOutput()

2.1: setOutput()

sd IPL

: User | CISCGUI | CISCSimulator | CPU | ROM

1: push IPL()

1.1: isPowerOff()

result

alt

[result==true]

1.2: initProcessor()

1.2.1: init()

1.2.2: setBootCode()()

1.2.2.1: getCode()

1.2.3: updateDisplay()

[else]

2: powerOff()

sd Memory load/store

CPU — Cache — Memory

1: loadMemory()

1.1: load()()

success or fail

alt

[result==fail]

1.2: load()()

success of fail

2: storeMemory()

2.1: store()

3: store()

sd Run

: User   CISCGUI   CISCSimulator   ControlUnit

1: [push Run]

1.1: isPoweroff()

1.2: loadUserInstruction()

1.3: run()

loop [state==READY]

1.3.1: singleStep()

sd Save

: User

CISCGUI

CISCSimulator

1: [push Save]

1.1: loadRegister()

sd Singlestep

: User   CISCGUI   CISCSimulator   CPU

1: [push Run]

1.1: isPoweroff()

1.2: loadUserInstruction()

1.3: singleStep()

loop [state==READY]

alt

[state==TERMINATE]

1.3.1: initProcessor()

1.3.2: clock()

result

1.3.3: isCurrentInstruction()

result

alt

[result==false]

1.3.4: [state = TERMINATE]

[else]

1.3.5: [state=READY]

1.3.2: updateDisplay()

sd TRAP/Fault

CPU

1: execute()()

1.1: setFault()()

1.2: setTrap()()

2: clock()()

alt

[MFR>=0]

2.1: get the address of machine fault code()

2.2: change PC()

alt

[trap>=0]

2.3: get the address of trap code()

2.4: change PC()