ESERCIZI in LABORATORIO – VERIFICA DEI MODULI PRINCIPALI DEL RISC-V (ESERCITAZIONE/LAB #04)

OBIETTIVI

1) **Capire la descrizione formale in Verilog di un semplice processore RISC-V**
2) **Capire come costruire il testbench per un modulo**

1) Capire la descrizione formale in Verilog di un semplice processore RISC-V

v. slide collegate https://arcal.dii.unisi.it/lab1/arcal_es04-verilog_riscv.pdf

2) Capire come costruire il testbench per un modulo

a) Scaricare i seguenti file (usando il link sottostante):

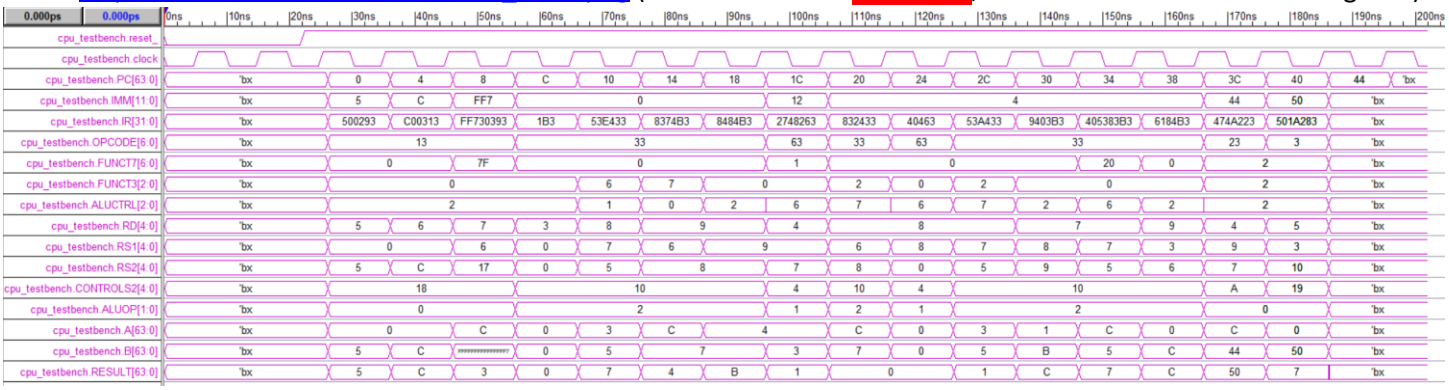
https://arcal.dii.unisi.it/tools1/rv64_test2.v (descrizione in Verilog di un semplice RISC-V)	https://arcal.dii.unisi.it/tools1/memfilerv.dat (codice macchina di un semplice programma)

```
module cpu_testbench;
   reg reset_; initial begin reset_=0; #22 reset_=1; #400; end
   reg clock;  initial clock=0; always #5 clock<=(!clock);
   wire[63:0] PC; wire[11:0] IMM; wire[31:0] IR; wire[6:0]OPCODE, FUNCT7;
   wire[2:0] FUNCT3,ALUCTRL; wire[4:0] RD, RS1, RS2; wire[1:0] ALUOP;
   assign PC=cpu.mycore.dp.pc,IR=cpu.mycore.dp.instr, ALUCTRL=cpu.mycore.dp.alu.aluctrl;
   ssign CONTROLS2=cpu.mycore.c.md.controls2, ALUOP=cpu.mycore.c.ad.aluop;
   assign OPCODE=cpu.mycore.c.opcode,FUNCT7=cpu.mycore.c.funct7;
   assign FUNCT3=cpu.mycore.c.funct3,RD=cpu.mycore.dp.rf.a3;
   assign RS1=cpu.mycore.dp.rf.a1,RS2=cpu.mycore.dp.rf.a2;
   wire[63:0] A,B,RESULT; assign A=cpu.mycore.dp.srca,B=cpu.mycore.dp.srcb;
   assign RESULT=cpu.mycore.dp.result, IMM=cpu.mycore.dp.gimm.y;
   initial begin wait(reset_); #180
      $display("X5=%h  (should contain '00000007')",cpu.mycore.dp.rf.rf[5]);
      $finish;
   end
   riscvmem cpu(clock,reset_);
endmodule
```

```
00500293
00c00313
ff730393
000001b3
0053e433
008374b3
008484b3
02748263
00832433
00040463
00000493
0053a433
009403b3
405383b3
006184b3
0474a223
0501a283
```

b) Generare un nuovo progetto Verilogger (v. esercitazione #01) e aggiungere il file Verilog rv64_test1.v

c) Provare a compilare ed eseguire la simulazione di questo codice: deve essere riprodotto il diagramma temporale https://arcal.dii.unisi.it/tools1/rv64_test2.png (nota: l'istruzione 00000493, dinamicamente non viene eseguita!)



d) Ad ogni gruppo verrà assegnato uno o più moduli (esempio "MAINDEC"). Il gruppo dovrà creare un nuovo progetto relativo solo a quel modulo (es. "MAINDEC") il cui codice può essere copiato dal file di cui sopra rv64_test2.v: scopo dell'esercizio e di scrivere il testbench specifico di quel modulo "isolato" in modo da testarne separatamente le funzionalità. A tal fine, il gruppo dovrà scrivere un testbench (seguendo il modello sottostante, disponibile al link https://arcal.dii.unisi.it/lab1/arcal_es04-rv64_testbench.v) da abbinare al modulo assegnato definendo i segnali di ingresso e i valori attesi sugli ingressi (nel testbench) e infine verificando che i segnali di uscita corrispondano a quelli del diagramma del punto c per quel modulo. La struttura del testbench sarà del tipo

```
module testbench;
   reg reset_; initial begin reset_=0; #22 reset_=1; #400; end
   reg clock;  initial clock=0; always #5 clock<=(!clock);
   reg …
   wire …
   initial begin wait(reset_)
    …
   $finish
   end
   XXX  xxx (clock,reset, <segnali_ingresso>,   <segnali_uscita>)
endmodule
```