

Design e implementazione di un ambiente di simulazione e testing in Verilog per il processore sEP8

Tesi di Laurea in Ingegneria Informatica

Candidato

Nicola
Ramacciotti

Relatori

Prof. Giovanni Stea
Dott. Raffaele Zippo



UNIVERSITÀ DI PISA

- Il sEP8 è un processore didattico a 8 bit
- Obiettivo
 - Rendere simulabile e osservabile il processore con un ambiente Verilog
- P. Corsini. *Dalle porte AND OR NOT al Sistema Calcolatore. Un viaggio nel mondo delle reti logiche in compagnia del linguaggio Verilog. Pisa: Edizione ETS, 2021*

- Completamento del sistema
 - Reti combinatorie (`valid_fetch`, `alu_result`, etc.)
 - Spazio di memoria e I/O
- Interazione simulazione-host tramite interfacce
 - Interfaccia di output che stampa a terminale il carattere ASCII ricevuto

■ Memoria ROM

```

1 main:
2     NOP
3     MOV $0x41,%AL
4     OR $0x20,%AL
5 loop:
6     OUT %AL,0x0000
7     ADD $0x01,%AL
8     CMP $0x7A,%AL
9     JBE loop
10    HLT

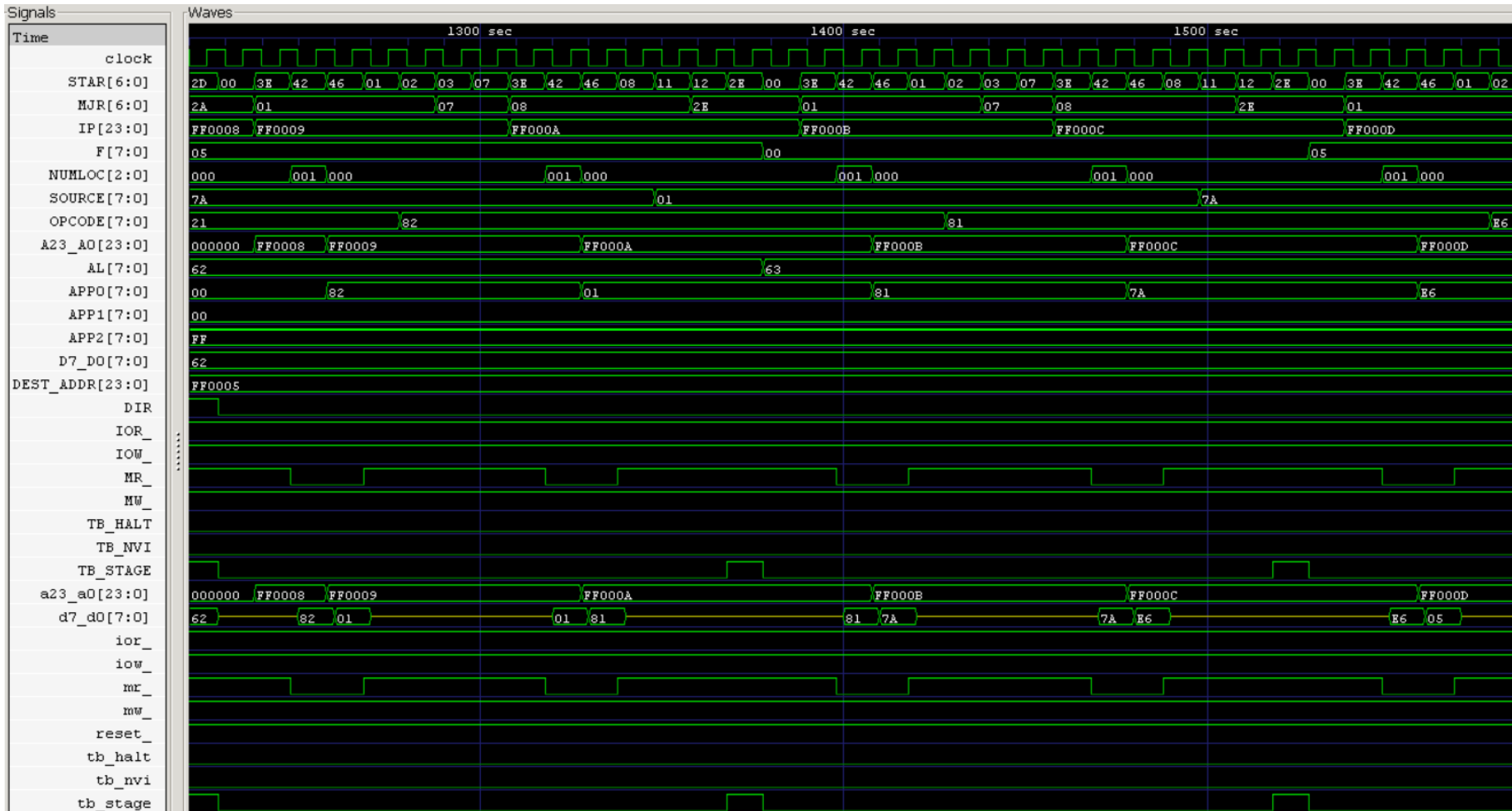
```

```

1 module ROM( a23_a0, s_, mr_, d7_d0 );
2     parameter ritardo_lettura = 2;
3     input [23:0] a23_a0;
4     input s_,mr_;
5     output [7:0] d7_d0;
6     `include "parametri_opcode.v"
7     function [7:0] valore;
8         input [23:0] a23_a0;
9         casex (a23_a0)
10             24'HFF0000: valore = opcode_nop; // NOP
11             24'HFF0001: valore = opcode_mov_operando_AL; // MOV $0x41,%AL
12             24'HFF0002: valore = 8'H41; // 'A'
13             24'HFF0003: valore = opcode_or_operando_AL; // OR $0x20,%AL
14             24'HFF0004: valore = 8'H20;
15             24'HFF0005: valore = opcode_outAL; // OUT %AL,0x0000
16             24'HFF0006: valore = 8'H00;
17             24'HFF0007: valore = 8'H00;
18             24'HFF0008: valore = opcode_add_operando_AL; // ADD $0x01,%AL
19             24'HFF0009: valore = 8'H01;
20             24'HFF000A: valore = opcode_cmp_operando_AL; // CMP $0x71,%AL
21             24'HFF000B: valore = 8'H7A; // 'z'
22             24'HFF000C: valore = opcode_jbe; // JBE loop
23             24'HFF000D: valore = 8'H05;
24             24'HFF000E: valore = 8'H00;
25             24'HFF000F: valore = 8'HFF;
26             24'HFF0010: valore = opcode_hlt; // HLT
27         endcase
28     endfunction
29     assign #ritardo_lettura d7_d0 = {s_,mr_}==2'b00 ? valore(a23_a0) : 8'HZZ;
30 endmodule

```

Esempio di evoluzione del processore



Base estensibile per lavori futuri

- Interfacce per input da tastiera, output a video
- Aggiunta di meccanismi più avanzati: interruzioni, cache, DMA, etc.



Conclusione

Grazie per l'attenzione