Laboration 5

Enchipsdator

Målet för denna laboration är att sätta ihop de förut skapade VHDL-modellerna till en enchipsdator. För att kunna styra och observera funktionen läggs en enkel monitorfunktion till.

1. Förberedelseuppgifter

Förberedelseuppgifter ska redovisas när laborationen börjar!

1. Lägg till en insignal STOP till Controllern från Laboration 4, som när den är aktiv (1) stoppar uppdateringen av nya tillstånd.
2. Skapa med strukturerad VHDL en modell för mikroprocessorn Electrum enligt Bilaga med tillägg av STOP-signal.
   Använd följande VHDL-skelett:

```
ENTITY CPU is
    PORT(  adr        : OUT address_bus;
           data       : IN instruction_bus;
           stop       : IN std_logic;        -- stops statemachine
           RWM_data   : INOUT data_bus;
           rw_RWM     : OUT std_logic;       -- read on high
           ROM_en     : OUT std_logic;       -- active low
           RWM_en     : OUT std_logic;       -- active low
           clk        : IN std_logic;
           reset      : IN std_logic);       -- active high
END ENTITY CPU;
ARCHITECTURE Structure of CPU is
-- osv
```

Figur 1. VHDL-skelett för mikroprocessorn

3. Skapa en modell för en enchipsdator enligt Bilaga.
   Till denna läggs insignalerna STOP enligt ovan, samt
   CHOICE som skall välja om man vill titta på adressbuss (0) eller databuss (1), respektive utsignalen s som visar adressbuss eller databuss.
   Eftersom CPU:n ”stannar” på adress 13 måste adressbussen tvingas till adress 15 när STOP är aktiv, för att man skall kunna konstatera att resultatet är korrekt.

Använd följande VHDL-skelett.

```
entity Enchip is
    port(clk     : in std_logic;
         reset   : in std_logic;  -- active high
         stop    : in std_logic;  -- stops statemachine
         choice  : in std_logic;  -- address(0) or data(1)
         s       : out std_logic_vector (3 DOWNTO 0)); -- output
end entity Enchip;
architecture Structure of Enchip is
-- osv
```

Figur 2. VHDL-skelett för enchipsdatorn

Programminnet (ROM) skall ges följande innehåll:

Adress (dec)	Innehåll
0	LDI R3, 3
1	STR R3, 14
2	LDI R1, 1
3	LDR R0, 14
4	MOV R0, R2
5	ADD R2, R1, R2
6	SUB R0, R1, R0
7	BRZ 12
8	NOP
9	BRA 5
12	STR R2, 15
13	BRA 13
övriga	NOP

Tabell 1. Program för test av mikroprocessorn

2. Arbetsgång

1. Simulera enchipsdatorn noggrant och verifiera att programmet exekveras korrekt. Härvid är det väsentligt att välja ut lämpliga signaler på olika nivåer för att visa funktionen.
2. Syntetisera enchipsdatorn och kontrollera att inga felmeddelanden finns.
3. Ladda ner enchipsdatorn på FPGA och testa funktionen.

3. Redovisning

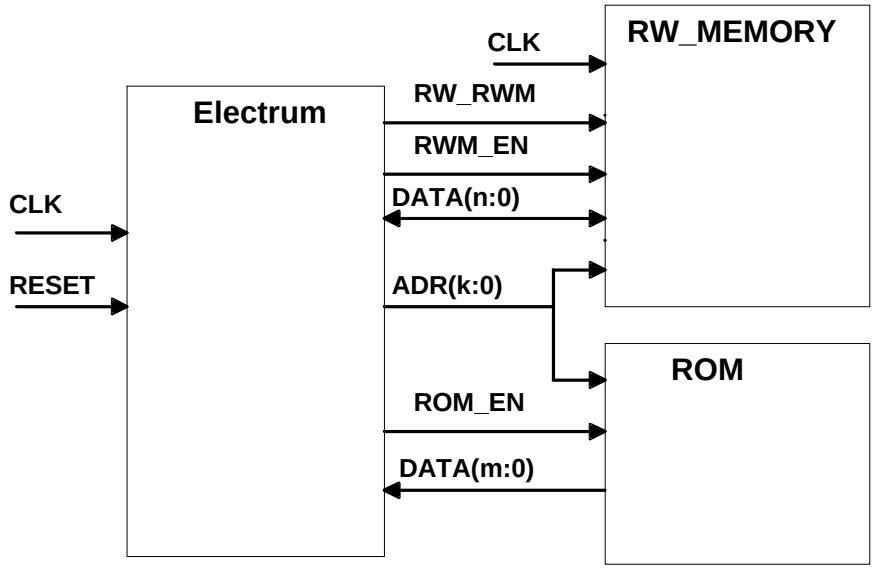
För en godkänd laboration ska följande visas:

- en väl dokumenterad enchipsdator-modell som uppfyller funktionskraven enligt Bilaga.
- Simuleringsresultat som visar att enchipsdatorn fungerar enligt specifikationen.

Bilaga
Mikroprocessor Electrum

Specifikation

Electrum är en enkel mikroprocessor av typ RISC (Reduced Instruction Set Computer). Tanken med mikroprocessorn Electrum är att den ska kunna anpassas efter användarens behov och det ska återspegla sig i VHDL-modellen. I grundvarianten är Electrum en 4-bitars processor med en adressbuss på 4 bitar, en instruktionsbuss på 10 bitar samt en databuss på 4 bitar. VHDL-modellen kan dock lätt anpassas till andra storlekar. Mikroprocessorn ansluts till två externa minnen enligt Figur .



Figur 3. Blockschema Electrum och dess anslutning till ett externa minnen