



Rys. 1 – Dane wyświetlane na ekranie monitora.

Opis wyświetlanych danych:

- PC – program counter,
- ADD – adres wyświetlany w czasie rzeczywistym,
- D11 – data_in – dane dostarczane do pamięci (cyfra 0 widoczna po „D11” to stan linii WE – write enable),
- D22 – data_out – dane dostarczane do procesora,
- ACC – zawartość akumulatora,
- ADD – adres, który jest obecnie wykorzystywany przez pamięć (synchroniczny)
- R0 – R3 – zawartości rejestrów R0, R1, R2, R3,
- C – flaga carry,
- 0 – flaga zera.

Wskaźnik koloru zielonego w obszarze pamięci oznacza operacje odczytu z komórki pamięci, a koloru czerwonego – zapis do komórki pamięci.

Obsługa środowiska testowego

Aby móc wyświetlać dane z procesora na monitorze VGA należy w pierwszej kolejności dodać do projektu moduł „mem2vga” zawarty w pliku o tej samej nazwie. Opis połączeń modułu przedstawiony jest na rysunku

```

module mem2vga (
    input memoryCLK, //zegar synchroniczny pamieci (aktywne zboczem narastajacym)
    input [7:0] DataIN, // Wejscie danych (zapis do pamieci)
    output reg [7:0] DataOUT, //Wyjscie danych (odczyt z pamieci)
    input [11:0] Adress, // Wybrany adres
    input WE, // Zapis (aktywny stanem wysokim)

    input pixelclk, //25,125mhz dla standardu 640x480@60Hz
    //      |       |       |       |       |
    //      |       |       |       |       | (wiekszosc monitorow przyjmuje z duza tolerancja)

    //wyjscia DSUB - VGA
    output hsync,
    output vsync,
    output r,g,b

    //dodatkowe debugi (rejestry procesora)
    ,input [11:0] PC
    ,input [7:0] R0
    ,input [7:0] R1
    ,input [7:0] R2
    ,input [7:0] R3
    ,input [7:0] acc
    ,input zero_flag
    ,input cy_flag
);

```

Rys. 3 – Moduł mem2vga.

Moduł korzysta z programu asemblerowym w pliku .hex. Przykładowy program w języku asembler przedstawiono na rysunku

```

ORG 0x0
MVI0 5
MVI1 10
MVI2 15
MVI3 20
DECR 0
RC 0
CC 0
MVA 0
SAVE 0xF0
ORR 1
SBBM 0xF0
LOAD 0xF0
ADCR 2
NOTR 3
ANDI 13
CMPM 0xF0
JZ 0x0

```

Rys. 4 – przykładowy program w języku asembler.

Powyższy kod po przekonwertowaniu do postaci hexadecymalnej, możliwej do odczytania przez moduł mem2vga znajduje się na rysunku poniżej.

8C 05
8D 0A
8E 0F
8F 14
98
DA
D6
80
70 F0
A9
F8 00 F0
60 F0
B2
93
C4 0D
FC 00 F0
30 00