



AGH

Sprawozdanie z laboratorium 1
Technika Mikroprocesorowa

Maciej Kmąk, Hubert Miklas

Wprowadzenie

Zadanie zostało zrealizowane najpierw w języku Python, a następnie przy użyciu układu FPGA “Zybo” produkcji Digilent w środowisku Vivado 2024.1 w języku opisu sprzętu (HDL) Verilog. Zadanie miało na celu zademonstrowanie rozwoju sprzętu typu RISC (Reduced instruction set computer).

Rozwiązania zadań

Zadanie 3.5 – Podstawowy program

Program na niezmodyfikowanym kodzie. Podany program wykonuje kolejno polecenia:

1. 80 (1000 0000) – Wpisz 0 do rejestru r0 – 100 kod instrukcji `mov reg, #imm;` – 0 – wskaźnik rejestru r0, 0000 wartość
2. 90 (1001 0000) – Wpisz 0 do rejestru r1 – 100 kod instrukcji `mov reg, #imm;`, 1 – wskaźnik rejestru r1, 0000 wartość
3. EE (1110 1110) – Wpisz wartość rejestru r0 na komórkę o wskaźniku 14 – 111 – kod instrukcji `str reg, addr;`, 0 – wskaźnik rejestru r0, 1110 – wskaźnik komórki 14
4. 86 (1000 0110) – Wpisz 6 do rejestru r0 – 100 kod instrukcji `mov reg, #imm;`, 0 – wskaźnik rejestru r0, 0110 wartość
5. DD (1101 1101) – Wybierz wartość z pamięci o wskaźniku 13 i wpisz ją do rejestru r0 – 110 kod instrukcji `ldr reg, addr;`, 1 – wskaźnik rejestru r1, 1101 – wskaźnik komórki pamięci 13
6. 20 (0010 0000) – Dodaj rejestry r0 i r1 i zapisz wynik w r0 – 0010 kod instrukcji `add r0, r1;` (0 może umownie wskazywać na rejestr r0, wtedy instrukcja 0011 mogłaby oznaczać tę samą operację tylko z wpisaniem wartości do rejestru r1), 0000 – nieużywane dane
7. EE (1110 1110) – Wpisz wartość rejestru r0 na komórkę o wskaźniku 14 – 111 kod instrukcji `str reg, addr;`, 0 wskaźnik rejestru r0, 1110 – wskaźnik komórki pamięci 14

c=0	r0=B	r1=5	pc=0	A=7	R=00	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=1	r0=B	r1=5	pc=0	A=0	R=00	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=2	r0=B	r1=5	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=3	r0=0	r1=5	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=1	r0=0	r1=0	pc=0	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=2	r0=0	r1=0	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=3	r0=0	r1=0	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=0	r0=0	r1=0	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=1	r0=0	r1=0	pc=0	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=2	r0=0	r1=0	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=3	r0=0	r1=0	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=0	r0=0	r1=0	pc=1	A=0	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=1	r0=0	r1=0	pc=1	A=1	R=80	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=2	r0=0	r1=0	pc=2	A=1	R=90	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=3	r0=0	r1=0	pc=2	A=1	R=90	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=0	r0=0	r1=0	pc=2	A=1	R=90	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=1	r0=0	r1=0	pc=2	A=2	R=90	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=2	r0=0	r1=0	pc=3	A=2	R=EE	W=0B	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=3	r0=0	r1=0	pc=3	A=E	R=EE	W=00	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=0	r0=0	r1=0	pc=3	A=E	R=0B	W=00	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	0B	00
c=1	r0=0	r1=0	pc=3	A=3	R=00	W=00	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	00	00
c=2	r0=0	r1=0	pc=4	A=3	R=86	W=00	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	00	00
c=3	r0=6	r1=0	pc=4	A=3	R=86	W=00	M:	80	90	EE	86	DD	20	EE	00	00	00	00	00	00	05	00	00
c=0	r0=6	r1=0	pc=4	A=3	R=86	W=00	M:	80	90	EE	86	DD	20	EE	00	00							

c=1 r0=0 r1=0 pc=0 A=0 R=00 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=0 r1=0 pc=1 A=0 R=80 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=0 r1=0 pc=1 A=0 R=80 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=0 r1=0 pc=1 A=0 R=80 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=1 r0=0 r1=0 pc=1 A=1 R=80 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=0 r1=0 pc=2 A=1 R=90 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=0 r1=0 pc=2 A=1 R=90 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=0 r1=0 pc=2 A=1 R=90 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=1 r0=0 r1=0 pc=2 A=2 R=90 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=0 r1=0 pc=3 A=2 R=EE W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=0 r1=0 pc=3 A=E R=EE W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=0 r1=0 pc=3 A=E R=0B W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=1 r0=0 r1=0 pc=3 A=3 R=00 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=0 r1=0 pc=4 A=3 R=86 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=6 r1=0 pc=4 A=3 R=86 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=6 r1=0 pc=4 A=3 R=86 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=1 r0=6 r1=0 pc=4 A=4 R=86 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=6 r1=0 pc=5 A=4 R=DD W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=6 r1=0 pc=5 A=D R=DD W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=6 r1=5 pc=5 A=D R=05 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=1 r0=6 r1=5 pc=5 A=5 R=05 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=6 r1=5 pc=6 A=5 R=20 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=B r1=5 pc=6 A=5 R=20 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=B r1=5 pc=6 A=5 R=20 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=1 r0=B r1=5 pc=6 A=6 R=20 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=B r1=5 pc=7 A=6 R=EE W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=B r1=5 pc=7 A=E R=EE W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=B r1=5 pc=7 A=E R=00 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=1 r0=B r1=5 pc=7 A=7 R=0B W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=2 r0=B r1=5 pc=8 A=7 R=00 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=3 r0=B r1=5 pc=0 A=7 R=00 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

c=0 r0=B r1=5 pc=0 A=7 R=00 W=0B M: 80 90 EE 86 DD 20 EE 00 00 00 00 00 00 05 0B 00

STR R0, [14]

pc update 2 -> 3

fetch opcode EE

mem address 2->E(14)

ex: write 00 to E(14)

MOV R0, #6

pc update 3 -> 4

fetch opcode 86

ex: value r0 0->6

LDR R1, [13]

pc update 4 -> 5

fetch opcode DD

mem address D(13) -> value 5

ex: value r1 0->5

ADD R0, R1

pc update 5 -> 6

ex: value r0 -> 5+6 = B

STR R0, [14]

pc update 6 -> 7

fetch opcode EE

mem address 6->E(14)

ex: write B to E(14)

JMP 0

pc update 7 -> 8

fetch opcode 00

ex: pc update 8->0

Zadanie 3.6 – Własna implementacja

Program na niezmodyfikowanym kodzie. Podany program wykonuje kolejno polecenia:

- 80 (1000 0000) – Wpisz 0 do rejestru r0 – 100 kod instrukcji mov reg, #imm; – 0 wskaźnik rejestru r0, 0000 – wartość
- 92 (1001 0010) – Wpisz 2 do rejestru r1 – 100 kod instrukcji mov reg, #imm; – 1 wskaźnik rejestru r1, 0010 – wartość
- 86 (10000 0110) – Wpisz 6 do rejestru r0 – 100 kod instrukcji mov reg, #imm; – 0 wskaźnik rejestru r0, 0110 – wartość
- 30 (0011 0000) – Dodaj rejestry r0 i r1 i zapisz wynik w r1 – 0011 kod instrukcji add r1, r0; 1 – wskaźnik rejestru r1, 0000 – nieużywane dane
- FE (1111 1110) – Wpisz wartość rejestru r1 na komórkę o wskaźniku 14 – 111 kod instrukcji str reg, addr; 1 – wskaźnik rejestru, 1110 – wskaźnik komórki pamięci 14
- 8D (1000 0110) – Wpisz 13 do rejestru 0 – 100 kod instrukcji mov reg, #imm; 0 – wskaźnik rejestru r0, 0110 – wartość
- 40 (0100 0000) – Wykonaj mnożenie $r0 * r1$ i 4 starsze bity wpisz do r0 młodsze do r1 – 0100 kod instrukcji mul r1, r0; 0000 – niewykorzystana wartość

```

86 2:
87 begin
88     state <= 3;
89     if(opcode[7]==0) //jesli bit 7 ma wartosc 0, to operacja arytmetyczna lub skok
90     begin
91         case(opcode[6:4]) //analizujemy bity 6, 5 i 4...
92         3'b000: //jesli wszystkie sa ustawione na 000...
93         begin
94             //wykonujemy skok pod adres dany stale natychmiastowa
95             pc <= opcode[3:0]; //zaladuj do PC wartosc znaleziona w bitach 3:0 kodu instrukcji
96         end
97
98         3'b010: //jesli bity 6:4 maja wartosc 010...
99         begin
100             r0 <= r0 + r1; //wykonujemy dodawanie: add r0, r1 (r0 = r0 + r1)
101         end
102
103         3'b011: //jesli bity 6:4 maja wartosc 011...
104         begin
105             r1 <= r1 + r0; //wykonujemy dodawanie: add r1, r0 (r1 = r1 + r0)
106         end
107
108         3'b100: //jesli bity 6:4 maja wartosc 100...
109         begin
110             r0 <= r0 * r1 / 16; // Wykonujemy mnozenie i zapisujemy starsze bity do r0
111             r1 <= r0 * r1 % 16; // Wykonujemy mnozenie i zapisujemy młodsze bity do r1
112         end
113     endcase
114 end

```

Rys. 2 – Implementacja wymaganych instrukcji

c=1 r0=0 r1=0 pc=0 A=0 R=00 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=2 r0=0 r1=0 pc=1 A=0 R=80 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=3 r0=0 r1=0 pc=1 A=0 R=80 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=0 r0=0 r1=0 pc=1 A=0 R=80 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=1 r0=0 r1=0 pc=1 A=1 R=80 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=2 r0=0 r1=0 pc=2 A=1 R=92 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=3 r0=0 r1=2 pc=2 A=1 R=92 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=0 r0=0 r1=2 pc=2 A=1 R=92 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=1 r0=0 r1=2 pc=2 A=2 R=92 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=2 r0=0 r1=2 pc=3 A=2 R=86 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=3 r0=6 r1=2 pc=3 A=2 R=86 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=0 r0=6 r1=2 pc=3 A=2 R=86 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=1 r0=6 r1=2 pc=3 A=3 R=86 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=2 r0=6 r1=2 pc=4 A=3 R=30 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=3 r0=6 r1=8 pc=4 A=3 R=30 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=0 r0=6 r1=8 pc=4 A=3 R=30 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=1 r0=6 r1=8 pc=4 A=4 R=30 W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=2 r0=6 r1=8 pc=5 A=4 R=FE W=00 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=3 r0=6 r1=8 pc=5 A=E R=FE W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=0 r0=6 r1=8 pc=5 A=E R=00 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 00

c=1 r0=6 r1=8 pc=5 A=5 R=08 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=2 r0=6 r1=8 pc=6 A=5 R=8D W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=3 r0=D r1=8 pc=6 A=5 R=8D W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=0 r0=D r1=8 pc=6 A=5 R=8D W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=1 r0=D r1=8 pc=6 A=6 R=8D W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=2 r0=D r1=8 pc=7 A=6 R=40 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=3 r0=6 r1=8 pc=7 A=6 R=40 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=0 r0=6 r1=8 pc=7 A=6 R=40 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=1 r0=6 r1=8 pc=7 A=7 R=40 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=2 r0=6 r1=8 pc=8 A=7 R=00 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=3 r0=6 r1=8 pc=0 A=7 R=00 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

c=0 r0=6 r1=8 pc=0 A=7 R=00 W=08 M: 80 92 86 30 FE 8D 40 00 00 00 00 00 00 00 00 08

MOV R1, #2

pc update 1 -> 2

fetch opcode 92

ex: value r1 0->2

MOV R0, #6

pc update 2 -> 3

fetch opcode 86

ex: value r0 0->6

ADD R1, R0

pc update 3 -> 4

ex: value r1 -> 2+6 = 8

STR R1, [14]

pc update 4 -> 5

fetch opcode FE

mem address 6->E(14)

ex: write 8 to E(14)

MOV R0, #D

pc update 5 -> 6

fetch opcode 8D

ex: value r0 6->D

MUL R1, R0

pc update 6 -> 7

ex: value r0 -> 6 ; r1->8

// D(13) * 8 = 104 = 0x68

JMP 0

pc update 7 -> 8

fetch opcode 00

ex: pc update 8->0