

Зміст

| | | |
|----------|---------------------------|----------|
| 1 | Практика | 2 |
| 1.1 | PDP | 2 |
| 1.1.1 | Алгоритм | 2 |
| 1.1.2 | Програмний код | 2 |
| 1.2 | Intel 8080 | 2 |
| 1.3 | MIPS | 3 |
| 1.3.1 | Функціонування | 3 |
| 1.3.2 | Практчні задачі | 4 |

Розділ 1

Практика

1.1 PDP

17.02.2014

Регістр R_4 . Його вміст має бути зменшений на 2. Отриманий результат є адрес, який потрібно отримати і занулити молодший розряд. І цей код записати у комірку пам'яті, що зсунута відносно номеру 155776 на -40 комірок.

1.1.1 Алгоритм

1. Зменшити значення регістру R_4 на 2;
2. Прочитати значення за адресою (R_4) і помістити його в R_3 ;
3. Накласти на значення, що міститься в R_3 маску, інвертовану відносно 111116;
4. Записати її у відповідну адресу.

1.1.2 Програмний код

```
SUB # 000002, R4;  
MOV (R4), R3;  
BIC # 000001, R3;  
MOV # 155776, R2;  
MOV R3, -40(R2);
```

1.2 Intel 8080

05.03.2014

Типи завдань:

- Призначення команди (інфіксна нотація, кількість циклів та інше);
- Функціонування та коментарі до нього;
- Швидкодія;

- Вміст регістрів та комірок пам'яті.

M - це завжди регістрова пара HL.

addr - це пряма адресація, адреса лежить в регістровій парі WZ.

Приклад 1. *CC addr C - це carry flag, в цей регістр буде поміщено ссув, який був отриманий у випадку арифметичних та інших операцій.*

1.3 MIPS

1.3.1 Функціонування

| |
|------------|
| 12.05.2014 |
|------------|

Що буде на контрольній роботі з MIPS:

- описание команды, обрамление, машинный код;
- описание алгоритма, комментарии к коду, (10 б);
- описать функционирование.

MIPS-lite інструкції, які можуть трапитися:

- ADD або SUB
- OR I
- LOAD або STORE Word
- BRANCH

Завдання 1.

$lw\$t1, offset(\$t2);$ (1.1)

Команда завантажує у регістр $\$t1$ дані з пам'яті за адресою $\$t2$ з сувом *offset*.

| № | Микрооперація | Управляющий сигнал |
|----|----------------------------------|---|
| 1 | PC_{out} | $IorD=0;$ |
| 2 | Цикл пам'яті; | $MemRead=1$ |
| 3 | $ALU_a := (PC)$ | $AluSrcA=0$ |
| 4 | $ALU_b := (4)$ | $AluSrcB=01$ |
| 5 | $ALU := ALU_a + ALU_b$ | $ALU_{op}=00 \rightarrow ALU_{control} = 0010 (= 2)$ |
| 6 | $ALU_{out} := (ALU)$ | $PCSource=00;$ |
| 7 | $PC := (ALU)$ | $PCWrite=1;$ |
| 8 | $IR := ((PC_{old}))$ | $IR_{wr}=1;$ |
| 9 | $CU[5-0] := IR[31-26]$ | |
| 10 | DC | |
| 11 | $A := IR[25-21]$ | |
| | $B := IR[20-16]$ | 9,10,11 пункт виконуються майже одночасно. Цими мікроопераціями мікропроцесор готує майбутню операцію R типу |
| 12 | $ALU_a := (PC_{new})$ | $ALU_{srcA} = 0$ |
| 13 | $ALU_b := (SE(IR[15:0]))? \ll 2$ | $ALU_{srcB} = 11$ |
| 14 | $ALU := (ALU_a) + (ALU_b)$ | |
| 15 | $ALU_{out} := (ALU)$ | $ALU_{op} = 00 \rightarrow ALU_{control} = 0010$ |
| 16 | $ALU_a := A$ | $ALU_{srcA} = 0$ |
| 17 | $ALU_b := SE(IR[15:0])$ | $ALU_{srcB} = 10$ |
| 18 | $ALU := (ALU_a) + (ALU_b)$ | $ALU_{op} = 00$ |
| 19 | $ALU_{out} := (ALU)$ | $ALU_{control} = 0010$ |
| 20 | $M_{address} := (ALU_{out})$ | $IorD = 1$ |
| 21 | ЦП | $MemRead = 1$ |
| 22 | $MDR := ((ALU_{out}))$ | $MemToReg = 1$ |
| 23 | $((IR[20-16])) := (MDR)$ | $RegWrite = 1$ |

1.3.2 Практичні задачі

Завдання 2. Поміняти місцем дві змінні $t1$ та $t2$ так, щоб не використовувалась додаткова пам'ять. Математичний алгоритм:

$$t1 = t1 + t2 \quad (1.2)$$

$$t2 = t1 - t2 \quad (1.3)$$

$$t1 = t1 - t2 \quad (1.4)$$

$$(1.5)$$

add \$t1,\$t1,\$t2

sub \$t2,\$t1,\$t2

sub \$t1,\$t1,\$t2

Завдання 3. Знайти кількість ненульових бітів у числі n .

```
addi $t1,$t1,0x49249249 # 0100 1001 0010 0100 1001 0010 0100 1001
addi $t2,$t2,0x381c0e07 # 0011 1000 0001 1100 0000 1110 0000 0111
addi $t3,$v0,0

and $t4,$t3,$t1

sra $t5,$t3,1
and $t5,$t5,$t1

sra $t6,$t3,2
and $t6,$t6,$t1

add $t3,$t4,$t5
add $t3,$t3,$t6

sra $t5,$t3,3
add $t4,$t3,$t5

and $t4,$t4,$t2

sra $t6,$t3,6
and $t6,$t6,$t2

add $t3,$t4,$t6

sra $t4,$t3,9
sra $t5,$t3,18
sra $t6,$t3,27

add $t4,$t3,$t4
add $t4,$t4,$t5
add $t4,$t4,$t6

and $t4,$t4,0x3f
```