

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт
до лабораторної роботи № 5
з дисципліни «Комп'ютерні системи»
на тему: «Дослідження програмної моделі RISC CPU»

Виконав:
ст.гр. КІ-38
Швець А.Ю.
Прийняв:
Козак Н. Б.

Львів 2022 р.

Мета: Навчитися здійснювати оцінку структури об'єкта (RISC CPU) на існуючій програмній моделі. Навчитись встановлювати структуру інтерфейсів об'єкта.

Теоретичні відомості:

RISC CPU це процесор, у якому реалізована архітектура із *скороченим набором команд* (RISC – Reduced Instruction Set Computer).

Головні зусилля у архітектурі RISC направлені на побудову максимально ефективного конвейєра команд, тобто такого, де всі команди вибираються із пам'яті і поступають у ЦП на обробку у вигляді рівномірного потоку, причому ні одна команда не повинна знаходитися у стані очікування, а ЦП повинен залишатися завантаженим на протязі усього часу.

Цю умову відносно просто можна реалізувати для етапу виборки. Необхідно лише, щоб всі команди мали стандартну довжину, яка дорівнює ширині шини даних, що з'єднує ЦП і пам'ять.

Крім однакової довжини команд, важливо мати відносно просту підсистему декодування і управління: складний пристрій управління (УУ) буде вносити додаткові затримки у формування сигналів управління. Шлях суттєвого спрощення УУ – скорочення числа команд, що входять до складу системи команд ЦП, форматів команд і даних, а також видів адресації.

Якщо підсумувати наведені вище та інші вимоги до архітектури із скороченим набором команд, то концепцію RISC-комп'ютера можна звести до наступних положень:

- виконання всіх (або, не менше, 75 % команд) за один цикл;
- стандартна, у одне слово, довжина всіх команд, яка дорівнює природній довжині слова і ширині шини даних і така, що допускає уніфіковану потокову обробку усіх команд;
- мале число команд (не більше 128);
- мала кількість форматів команд (не більше 4);
- мале число способів адресації (не більше 4);
- доступ до пам'яті тільки через команди „Читання” і „Запис”;

- всі команди, за виключенням „Читання” і „Запис”, використовують внутрішньопроекторні між регістрові пересилання;
- пристрій управління „жорсткою” логікою;
- відносно великий (не менше 30) процесорний файл регістрів загального призначення (як відомо, у сучасних RISC CPU число РОН може перевищувати 500).

Завдання:

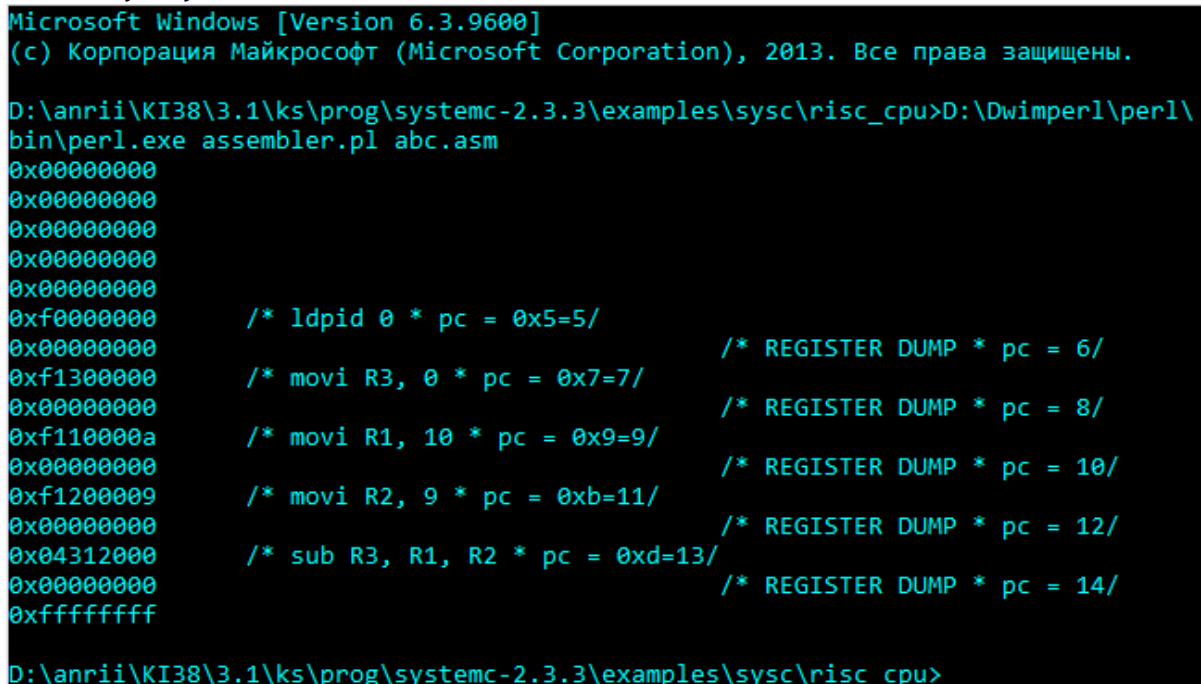
1. Дослідити програмну модель RISC CPU;
2. Визначити склад програмної моделі RISC CPU.
3. Визначити призначення блоків у структурі RISC CPU.
4. Визначити зв'язки між структурними блоками RISC CPU (інтерфейси).
5. Визначити структури інтерфейсів між блоками RISC CPU.
6. Визначити окремі потоки у структурі інтерфейсів:
 - інформаційні;
 - керування.

Хід виконання роботи:

Нехай $R1 = 10$, $R2 = 9$, $R3=10-9$

Abc.asm

```
ldpid 0
movi R3, 0
movi R1, 10
movi R2, 9
sub R3, R1, R2
```



```
Microsoft Windows [Version 6.3.9600]
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.

D:\anrii\KI38\3.1\ks\prog\systemc-2.3.3\examples\sysc\risc_cpu>D:\Dwimper1\perl\
bin\perl.exe assembler.pl abc.asm
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000
0xf0000000 /* ldpid 0 * pc = 0x5=5/
0x00000000 /* REGISTER DUMP * pc = 6/
0xf1300000 /* movi R3, 0 * pc = 0x7=7/
0x00000000 /* REGISTER DUMP * pc = 8/
0xf110000a /* movi R1, 10 * pc = 0x9=9/
0x00000000 /* REGISTER DUMP * pc = 10/
0xf1200009 /* movi R2, 9 * pc = 0xb=11/
0x00000000 /* REGISTER DUMP * pc = 12/
0x04312000 /* sub R3, R1, R2 * pc = 0xd=13/
0x00000000 /* REGISTER DUMP * pc = 14/
0xffffffff

D:\anrii\KI38\3.1\ks\prog\systemc-2.3.3\examples\sysc\risc_cpu>
```

віднімання чисел за допомогою perl

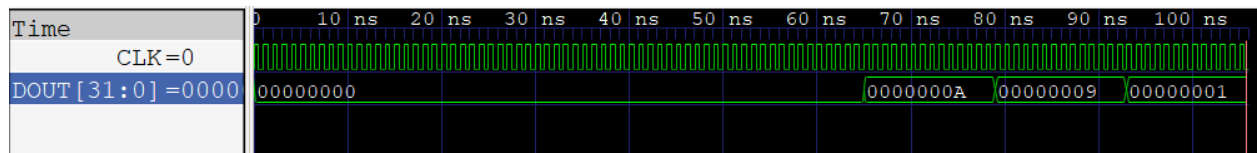
////////////////////////////////////

Time for simulation = 1

```
DOUT [ 3 ] = 0
DOUT [ 2 ] = 0
DOUT [ 1 ] = 0
DOUT [ 0 ] = 1
```

```
0x04312000
```

додавання значень до файлу icache



Часова діаграма проекту

Висновок: Навчився здійснювати оцінку структури об'єкта (RISC CPU) на існуючій програмній моделі. Навчився встановлювати структуру інтерфейсів об'єкта.