Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №6

з дисципліни: “Комп’ютерні системи”

на тему: “Побудова функціональної схеми процесорів RISC CPU”

Виконав: ст. гр. КІ-32

Кононенко Р.Ю.

Прийняв:

Козак Н.Б.

Львів – 2020

**Мета роботи:** навчитись розробляти нові функціональні модулі з метою розширення функціонування базової архітектури процесорів RISC, що відповідає програмній моделі, а також створення функціональної схеми.

**Індивідуальне завдання:** внести необхідні зміни в проект з 5-ї лабораторної (типи кешів, розмірності шин даних, назви та типи сигналів, операції, буферні модулі і т.д.) з метою реалізації функціоналу обраного процесору.

**Завдання:** Hewlett-Packard's PA-RISC, also known as HP-PA.

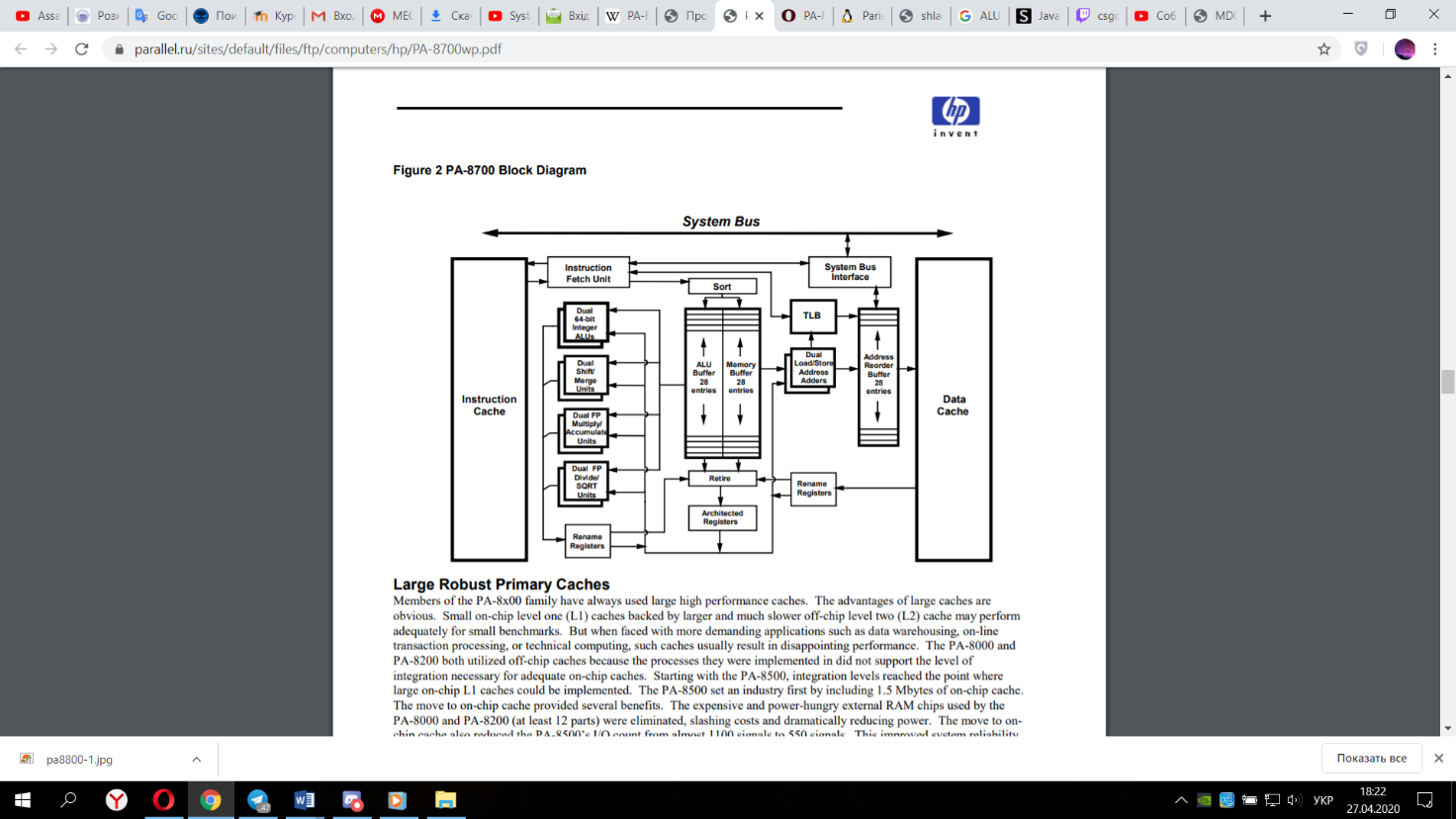
**Теоретична частина**

PA-RISC - мікропроцесорна архітектура, розробка в компанії Hewlett-Packard під керівництвом Рубі Б. Лі. Як слідує з назвами, це RISC-архітектура; PA значить Precision Architecture (Точна архітектура). Інноваційна фаза архітектури називається HP / PA (Hewlett Packard Precision Architecture).

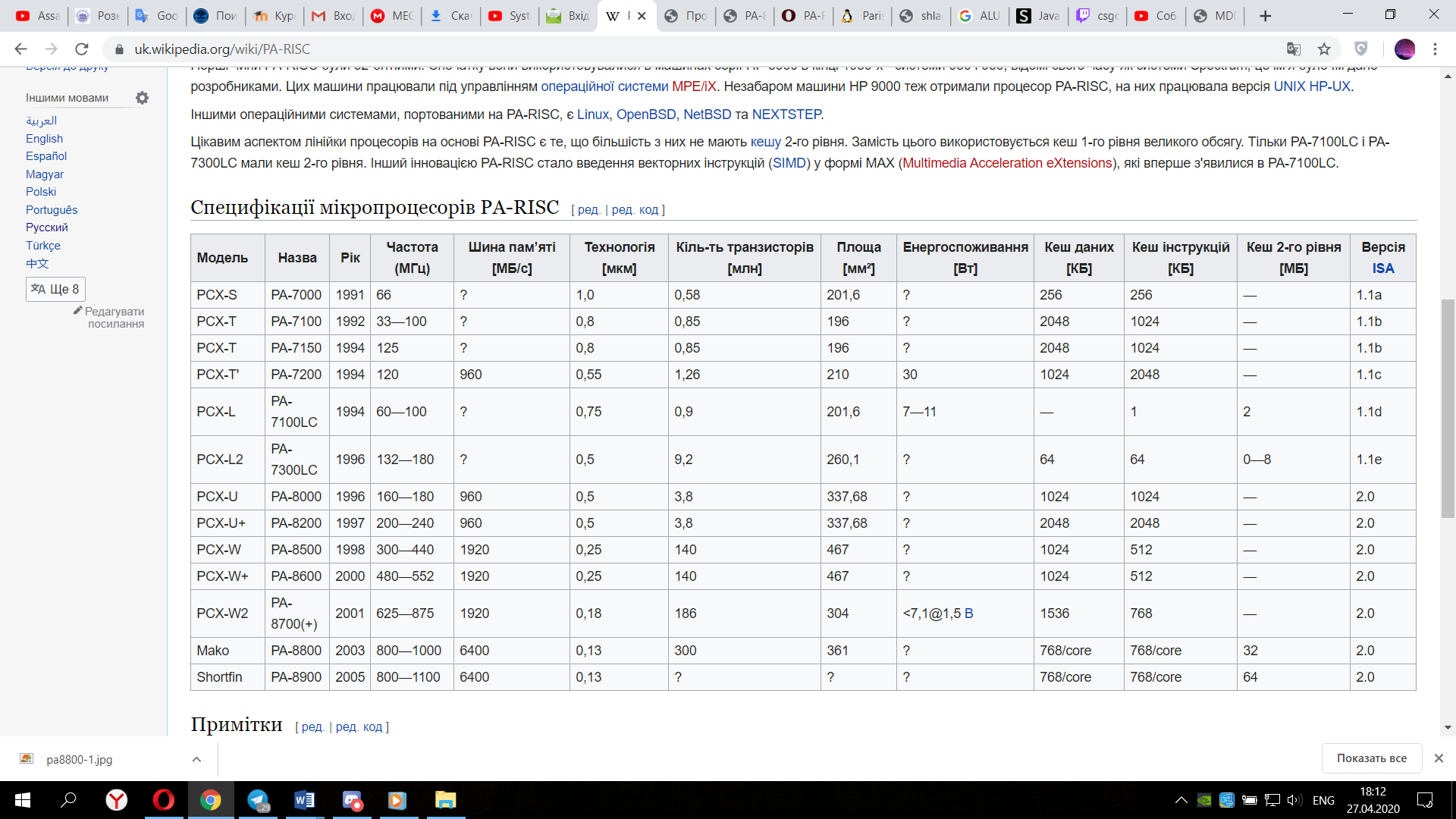
Перші процесори PA-RISC були розроблені та використані в середині-кінці 1980-х на ранніх серверах HP 9000/800 та системах HP 3000 MPE / iX. Точна схема іменування не є повністю зрозумілою, оскільки деякі джерела називають ці процесори як TS-1, NS-1 і NS-2, а інші називають такі ж процесори PN-5, PN-7 і PN-10. Ці ранні процесори все ще в основному були багаточіповими процесорами з окремими мікросхемами та компонентами, що утворюють центральний блок обробки, всупереч здебільшого одночіповим реалізаціям після PA-7000. Фішки були засновані на виробництві TTL, потім NMOS-III і нарешті CMOS26B. Цікавим аспектом цих процесорів є їх величезний TLB - від 2048 до 16384 записів, тоді як їхні наступники та конкуренти мали розміри, як правило, до кількох сотень.

Еталонною моделлю був обраний PA 8700 Processor (посилання на [datasheet](https://parallel.ru/sites/default/files/ftp/computers/hp/PA-8700wp.pdf)),

Для пояснення внесених змін у програмну модель RISC CPU наведена блок-діаграма вищезгаданого процесора



*Рис.1 Структура процесора HP-PA 8700*



*Рис.2 Мікропроцесори архітектури HP-PA*

Отже, судячи з рис.1, програмна модель RISC CPU не потребує створення додаткових буферних або програмних блоків, а також видалення вже існуючих. Зміни стосуватимуться лише розрядності шин адрес/даних, а також додавання відповідних команд, яких не вистачає.

1. Модифікував програмну модель RISC CPU згідно індивідуального завдання:

**Лістинг модифікованої програмної моделі:**

*bios.h*

Збільшив довжину вхідних файлів до .75 Мб (786432 байт), що відповідає функціоналу PA-архітектури

//..

imemory = new unsigned[786432];

itagmemory = new unsigned[786432];

printf("\*\* ALERT \*\* BIOS: initialize BIOS\n");

for (size = 0; size < 786432; size++) { // initialize bad data

imemory[size] = 0xffffffff;

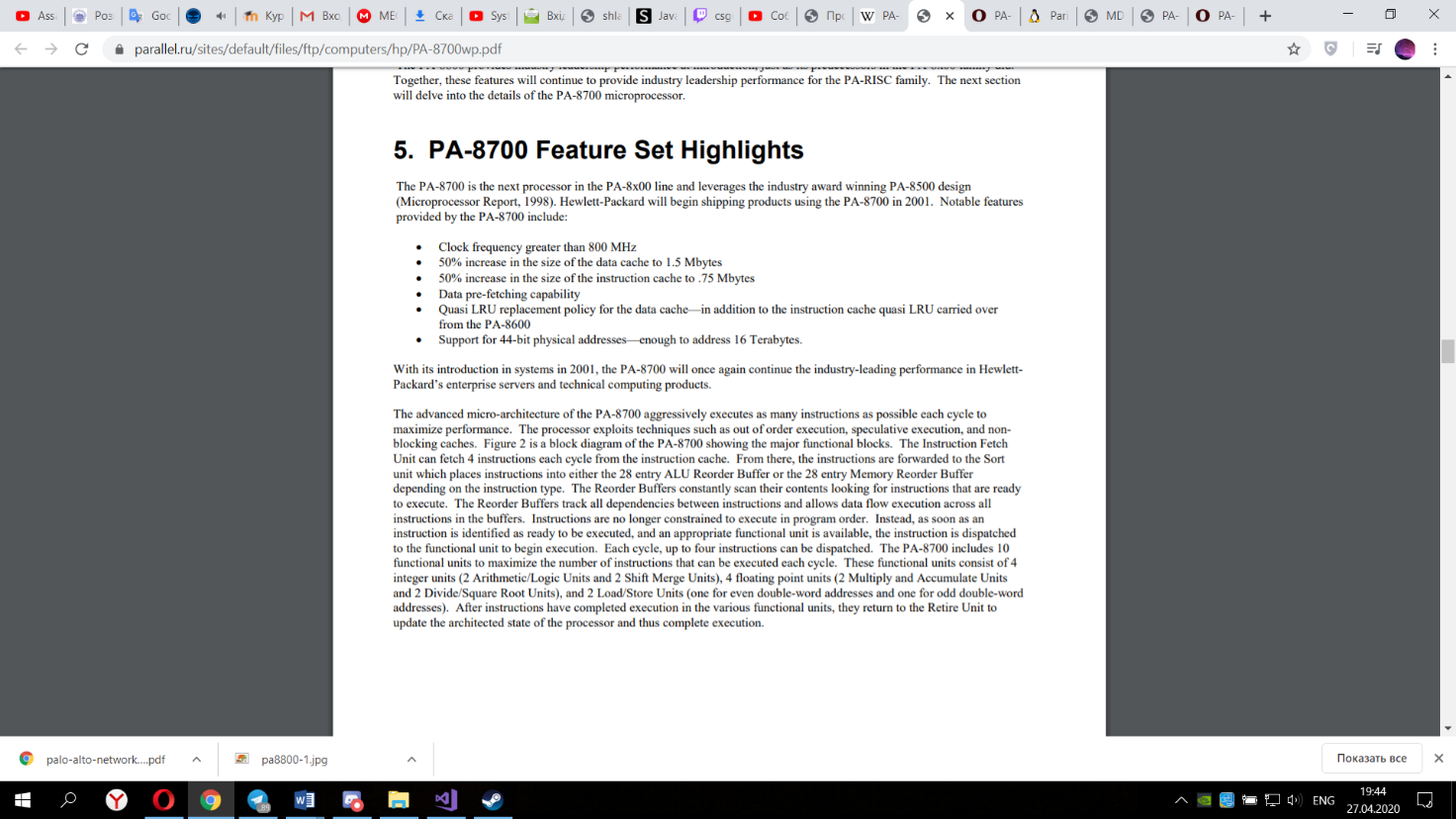
itagmemory[size] = 0xffffffff;

}

//..

Розширив шину адрес до 64 розрядів, що відповідає функціоналу PA-архітектури

#define BOOT\_LENGTH 64

Для підтвердження правильності внесених змін нижче наведені характеристики буферних блоків інструкцій та даних процесора HP-PA 8700

*Рис.3. Атрибути буферу інструкцій та даних процесора HP-PA 8700*

*icache.h*

Аналогічно розширив шину адрес та змінив довжину кодового слова операції в блоці кешу інструкцій згідно функціоналу РА-архітектури

#define BOOT\_LENGTH 64

#define MAX\_CODE\_LENGTH 800

*dcache.h*

Аналогічні модифікації довжини вхідного файлу згідно з файлом *bios.h* та *icache.h*

//..

dmemory = new unsigned[1572864];

dsmemory = new unsigned[1572864];

dtagmemory = new unsigned[1572864];

printf("\*\* ALERT \*\* DCU: initialize Data Cache\n");

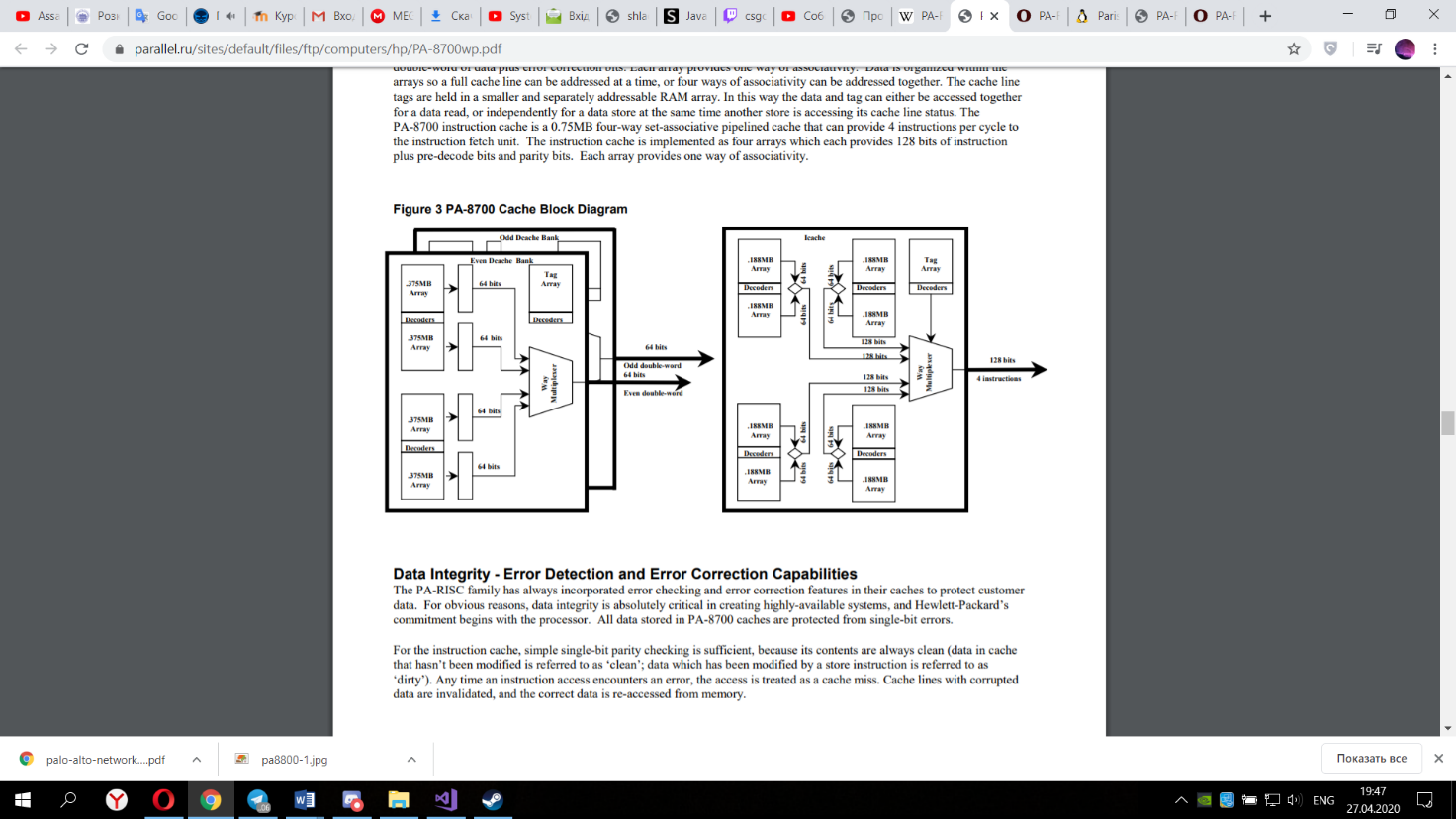
//..

for (i=size; i<1572864; i++) {

dtagmemory[i] = 0xdeadbeef;

dmemory[i] = 0xdeadbeef;

dsmemory[i] = 0; }}

//..

*Рис.4. Розрядність шин процесора HP-PA 8700*

*exec.cpp*

Розширив кількість операцій в блоці обчислень: додана операція ділення(division, усі інші операції аналогічні MIPS-архітектурі)

//..

case 15: // shladd

dout\_tmp = dina\_tmp << add1\_tmp + dinb\_tmp;

wait();

break;

//..

case 16: // UXOR

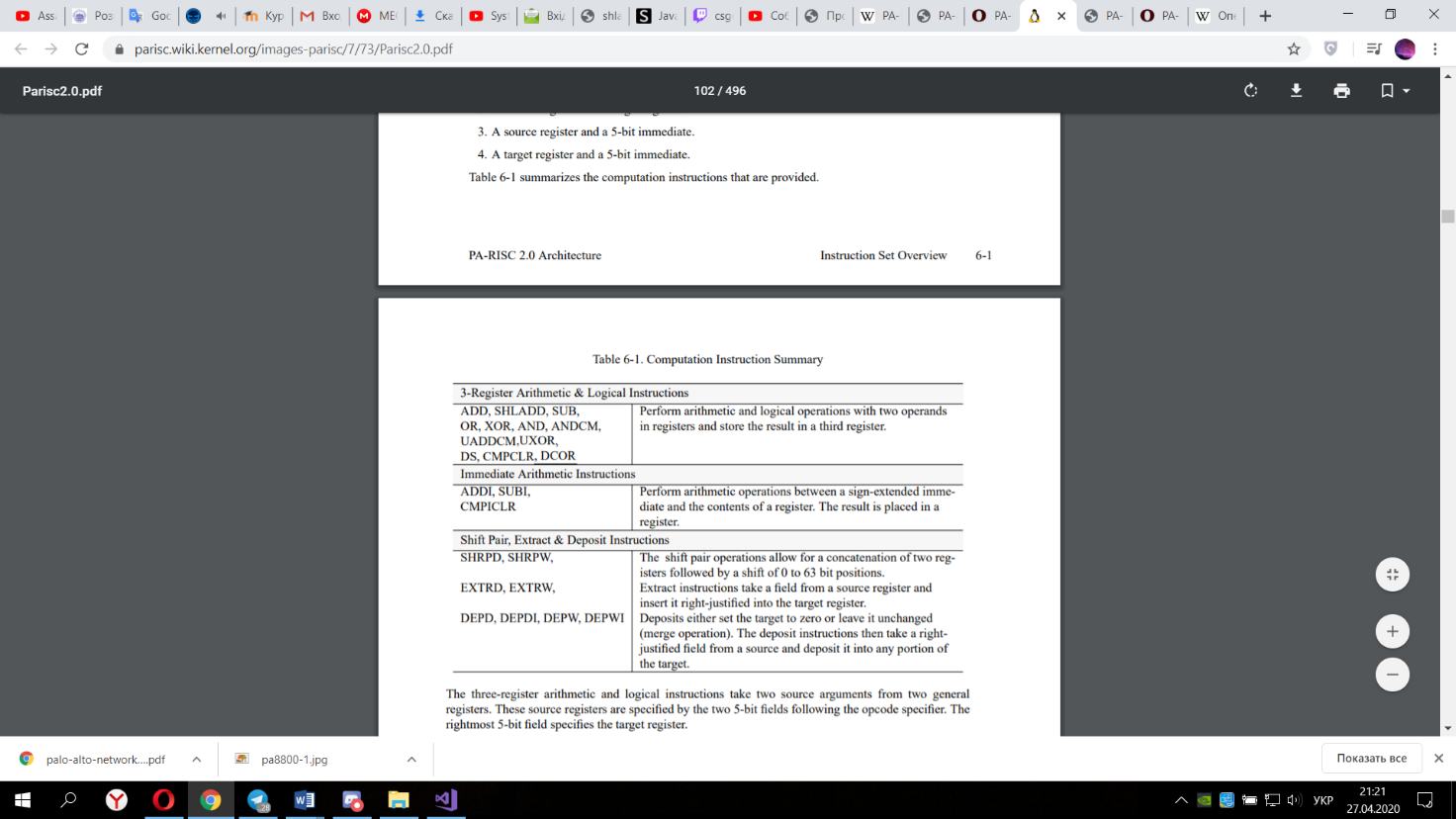
if (dina\_tmp < 0) dina\_tmp = -dina\_tmp;

if (dina\_tmp < 0) dinb\_tmp = -dinb\_tmp;

dout\_tmp = dina\_tmp ^ dinb\_tmp;

wait();

//..

Для підтвердження правильності внесених змін нижче наведені характеристики та операції блоку обчислень процесора HP-PA 8700

*Рис.5. Характеристики та операції блоку обчислень процесора HP-PA 8700*

*decode.h*

//..

signed int cpu\_reg[64]; //CPU register

signed int vcpu\_reg[64]; //virtual CPU register

bool cpu\_reg\_lock[64]; //lock architectural state register

unsigned int pc\_reg; //pc register

//..

for (int j =0; j<64; j++) vcpu\_reg[j] = 0;

for (int k =0; k<64; k++) cpu\_reg\_lock[k] = 0;

//..

Усі інші файли даної програмної моделі не потребують модифікації, оскільки повністю збігаються із MIPS-архітектурою, яка є індивідуальним завданням даної лабораторної роботи.

Багато представлених команд (деякі не вносились в роботу) можуть бути використані лише при написанні програмного коду мовою Assembler, тому їх реалізація у заданій програмній моделі не розглядається.

1. Створив функціональну схему модифікованої програмної моделі згідно індивідуального завдання:



*Рис.5. Функціональна схема модифікованої програмної моделі*

*згідно MIPS-архітектури*

Модифікована функціональна схема за своєю структурою не відрізняється від вихідної програмної моделі, бо в ході роботи даної лабораторної роботи було вказано, що жодні програмні блоки не були видалені або додані. Усі внесені зміни позначені зеленим кольором і полягають у зміні розрядності відповідних регістрів та шин адрес/даних згідно модифікованої програмної моделі, тобто змін у лістингу програми, що наведені вище.

**Висновок:**

На даній лабораторній роботі навчився розробляти нові функціональні модулі з метою розширення функціонування базової архітектури процесорів RISC, а також створив функціональну схему модифікованої програмної моделі.