Next Thursday HW #2 Due

Last Tuesday we talked abut 3 instruction det Arch

1. ARM \_ RISC

2. 68,000 \_ CISC  
3. Pentium \_ CISC

68,00

C EQU $ 20220

ORG

3) CISC- PENTIUM

BEGIN WITH ASSEMBLER DIRECTIVE{

DATA

NUM DD 17, 3, -51, 242, -113

N DD 5

SUM 0

CODE

Main: LEA EBX (E signifies 32 bits and x signifies 16 bits)

EDI MOV ECS, N counter

MOV EAX< O  
MOV EDI, 0

ENC EDI //increment

DEC ECX //decrement

Startadd:

ADD EAX[EBX + EDI 4]

INC EDI

DEC ECX

JC STANDARD

MOV SUM, EAX

END MAIN

COD IS IN BOOK CHAPTER 3. WE ARE NOT TESTED ON SYNTAX

EACH NUMBER TAKES 32 BITS

What are /were the similarities between the 3 instruction sets?

A) Data section { } (called directives which tells us where to store the data)

Code { }

B)

What was different?  
-syntax opcode src, dst, dst, src

-RISC, (Small number of registers) CISC more

-assembler directives (the specifics were different and not the concepts)  
# of registers/

-Addressing made specifics were different

Some may allow only certain post-increment or postincremen \_+ \_-

Today: we want to concentrate on Coldfire \_

Is port of M6800 family CISC

Opcode src, dst

Risc computer became popular because of pipelining.

Coldfire has only a subset of instructions from M68000

It is a variant of M68000 where there are less instructions available

The Goal is -> make it look more like a RISC machine BUT IT IS NOT A RISC MACHINE

They tried to simplify and keep the important instructions and removed some of the more complex instructions

Therefore, a colfire code should work on a M68000 but an M68K does not necessarily work on coldfire

Note that your book has appedite

In Coldfire- Most Arithmetic and logic instructions can act on long words only

The instructions that decrease W, L, S, CLR, MOVe,

After they removed some instructions they realized that they should reinstate some of the such as CMP, CMPI  
(combe and now with tby the word as loop)  
  
Furthermore remember that 68K CISC could have variable length instructions in cold fire they restricted them to not be longer than 6 bytes

Also certain addressing modes very restrictive for MOVE M, you can’t have post/pre decrement

Coldfire has only one stack pointer

-And certain things are not supported at all

Example BCD-

In Coldfire, contain instructions cannot act directly on memory

Therefore the estiniation must be a register

Ex:

ADDI LSL (left)

AND I LSR (right)

COMP I (Shift)

ORI, SUBI, NOT, NEG

ColdFire Manual read the first section

You need a data register or address register

Data registers use D

Address registers use A

Go to table 3.1

MOVE Moves immediate value to data register

MOVEA Move to address register

NEG

ADD Source + Destination -> Destination

ADD.L <ea> y, Dx

Dx is destination while y is source

No address register but simply data register

ADDA <ea> (stands for effective address)

If <ea> is #<data> then it is an immediate value

(Ay)+ would be post increment

ADDI (Add Immediate)

On Tuesday we start chapter 4

On Thursday HW#2 is Due

HW #3 is optional

Problems at chapter 3 coding (this counts as extra credit)

Due by the end of the term. If you do it yourself you get more credit.

Midterm March 7th

So far:

HW Chapter 1 Basic Organization of Computers

Group HW now HW Chapter 2 Instruction set

Chapter 3 (5th edition)

HW #3 for extracredit

In 6th edition -> Appendix Coldfire © ARM (D) INTEL (E)  
  
Starting next HW, I will assign smaller HW to be don3e i3ndividualy

(but you’ll have the option at wanting in groups if you like to individually

The topics we will focus from here one:  
1. I/O

2. Memory

3. Arithmetic 1-5 should take at least one week

4. Basic Processing Unit

5. Pipelining

6. Advance topics (last 1-2 weeks)

a) Embedded Systems

b) System on chips

c) Parallel processing

5th edition 6th Edition

From here onwards after reading Chapter 1 and 2 just concentrate on notes

**Bus**- What is a bus? (I/O organization section) +( Interconnection section)  
  
A Bus provides a communication path between two or more devices that are connected to it

Basic assumptions for a bus are:

1) only one communication at a time.

2) Each communication takes O(1) (constant) time to travel

Communication is established by

1) Broadcasting a message

|destination address| datas2w|

2) All ports will hear the broadcast but only one (i.e. the matching the address)

Responds or receives.

Example: Given n Function units THAT ARE CONNECTED WITH A BUS, SHOW HOW TO ADD N NUMBERS WHERE EACH OF THE NUMBERS IS IN THE REGISTER unit of one functional unit also assume they are linearly connected as well

Size of group is n^(1/2)

Given n units

Speed up = n/(n^(1/2)) = n^(1/2)

1. Don’t wear sunglasses

2. Like Professor Ung, tell jokes

3. Hate Coldfire

4. Too many problems (extra credit, option to work in groups

5. Quizzes (some like, some don’t)

We can use two busses. One carries address and the other carries the data.

Application of BUS for I/O design

Components of computer, Memory, processor, I/O

We can connect these components with a bus

How are I/O access handled?

Typically:  
An I/O device issue an interrupt request that signals the processor to announce that it needs an I/O process execution

There are 4 responses to the I/O request

1. The processor does not listen to request. (this happens if interrupts are disabled)
2. Interrupts are not disabled. It listens but it ignores it because it is doing a “higher” priority/task
3. Not disabled. It decides to process it
4. The interrupt is of a higher priority than its task
5. It always handles any interrupts
6. Not disabled. It asks the DMA (Direct Memory Access) to handle it.

What is DMA?

Assistant to the processor.

Condition: only if it requesting memory.

Another option is:  
Use a secondary processor to handle the interrupts if the primary processor is busy.

How to process an interrupt?  
(either by primary or secondary processor)  
-Save the program counter

-Save intermediate results

-put thing on hold

-Go and handle the interrupt

-Come back when done

\*status: busy :1

-resume operations

What happens when there are multiple intercept requests?  
1) If you get multiple requests and you queue them. Not simultaneous on a single bus)

2) You get a simultaneous either on a single bus or multiple buses. Multiple buses you can que them

Single bus use switches that connect or disconnect to I/O devices

3) Priority Encoder (Decoder)

EE 357 Notes

Everything covered up until now will be included on the Exam

Ask after class what chapters.

The exam will be the most similar to the sample exam online

Clock cycle: Nanoseconds

How do you know what you will use a lot

I just used it so I’ll be using it again

I just used it so I won’t be using it again recently

Two Types

1) Temporal locality deals with the concept of time

2) Special what memory locations will I use

While you’re at the store to get bread you see eggs and realize you’ll need it later so you get it

Same concept for computers

The Hierarchy of Memory from Fastest to longest (smallest to biggest also)

Register, Cache, Main Memory, Secondary Memory

Today: Memory

Processor, Cache, Main Memory, Disk,

One or more processors, Cache is multi-layer cache, Memory Module

Virtual Memory-is a memory management technique developed for multitasking kernels (WIKI)

Main Memory is much more smaller.

Virtual Memory is real it is just that only a subset of the physical memory resides in the main memory

1) Virtual Memory is a subset of physical secondary memory which is main memory.

2) Multiple process could address the same location by different (virtual address).

Design issues-

1) What to place in higher memory. Things that you need to use a lot

Things that you need to use

Problem: We don’t have enough space to bring all the things we may need

One solution: “Be selective” using concepts of locality

a) Temporal (Least frequently used, most frequently used, most recent, least recent, FIFO, LIFO

b) Spatial-particular location in memory block

REPLACEMENT

2) What to move out

3) What size of Data to place /replace (should you take one element or a chunk?) It really depends (one element, block, page, segment)\

4) How often should you update the memory?  
-after every write?  
-after certain period time? Depends

Processor Caches, main, secondary,

A user asks for a memory location by a virtual address

These address needs to be translated into a physical address

Translation takes time-> so we keep a translation table

Translation table

Has virtual addresses that corresponds with a physical address

Use the concept of locality to figure out which pair to replace with

TO MAP FROM MAIN MEMORY TO CACHE, USE FIXED MAPPING TECHNIQUES SUCH AS

1)Direct Mapping ( the ith member of the main memory can only go to a specific location in the Cache)

2) Fully Associative- can go to anywhere in cache

3) Set associative- to a specific set but anywhere in the set

Memory: Cache, Main, secondary

I/O: Interrupts, I/O processing (polling, daisy), DMA

Processor : Control Unit, ALU, Instruction register

Steps for executing an instruction?

Instruction Fetch

Instruction Decode

Operand Fetch

Execute

Write Back 5 Set pipeline (use RISC)

T=1 i1, iF

T =2 i2, iF i1, iD

T=3 i3,IF i2, iD i1,of

T=4 i4,if i3, iD i2,OF, i11,Ex

If you have multiple ALUs?

Put them in separate pipelines

Multiple executions at any time

Example:  
i1 A = B+C Program counter pointing to first instruction

I2 D = A +F Skip this step because we need to compute A first

I3 C = B+K

I4 J = L+M This and next step is called write after write

I5 J = P+Z Read after write

Read after write Cannot ignore

Write after read If you do it (concurrently) at the same time it should be okay

Write after write

Idea of “keep going” -> out of order execution

You can with the last one if no other dependencies

J1 and J2 register renaming

1st cycle i1 A = B+C

I3 C = B+K

I5 J = P

This Thursday Chapter 1-9

80% of exam

Exam will be smaller subset of this practice midterm

READ CHAPTER 1 and 2

Design a computer with the following specifications

1. With a RISC processor which has two Functional Units

a) 1 Adder/Subtractor – Ripple carry adder

b) Adder/Subtractor – Carry look ahead adder

Make it into a subtractor with XOR gate the top

You either generate a carry or propagate a carry when cin = 1 and at least a or b is 1

Cout = g + p

ai bi + Cin –(ai+bi)

Still takes O(logn) time

2) Show the memory organization such that Memory organization such that it has

-a 2 layer cache

-Main Memory with multiple modules

-virtual memory

(need to specify the size and speed)

Cache speed is like 10ns but 500ns is NO GOOD

Main Memory 200ns

Put sizes that are relative to your computer

Use off the shelf components

Google the appropriate memory sizes

Main memory: MM1 (Memory module 1)

How to connect or select them?

Put a mux to select the memory

Or use a decoder

What if you want to access a particular memory. And in that memory a specific address?

Say 8 Modules

And 1000 elements

3 bits which module and 10 bits for 1000 elements in a string

Explain placement (FIFO, Most recently used, Rplacement, mapping, cache-main memory, direct, set associative, fully associated, read, write, update.

ALL OF THIS CAN BE FOUND IN THE BOOK!

Proc. Smallest cache is fastest usually

3) Show the I/O organization

-DMA (direct memory access)  
-How one or more interrupts are handled

-Priority encoder

-Can you use a mux instead of a priority encoder? Yes but priority encoder is better

-polling/daisy chaining

-single mutiple bus

-FIFO, queue

You have to use select bits to select the bus

Review bus write and read

Select your own design

But for exam you need to know all!

20% of what you see here.

Daisy chaining- process the first and then the next and then the next (it means processing things one at a time)

Daisy chaining use concept of polling

Polling: you don’t know the ID so you ask if you need something. You ask them instead of them asking you

There are three more section

#4) Design your own very basic instruction set ( You come up with your own syntax (AWW YEAH!)

Add destination, source = add destination + source = destination

Assume that there is no instruction

For Multiply/Divide

Write a code using your own syntax Multiply and Divide

Such that all inputs and outputs are in 2’s complement format and can be positive or negative

What does this mean?

Add or subtract number using loop

Account for multiplying/dividing + or – numbers

Take track of overflow in

Op code, source, destination

What is distinguishing factor between RISC and CISC

RISC has fewer instructions (Load,

CISC has more instructions

5) Show how instructions are executed in your computer in a pipeline fashion. IF, ID, OF, EX, WB (Write Back)

Should you pre-fetch instructions? Put them the cache or instruction register

Make sure the registers, P.C. is in your processor section

6) To demonstrate the features of your design, show how out of order execution of a simple program can take place

Super scalar piplined RISC processor

Give an example

Program can be in C++

In you use C then explain how it is translated into assembly and machine binary codes

C -> Compiler -> Assembly -> Assembler -> Binary code -> executed

Cascade Decoders

Out of order operation means that operations are done out of order to see which ones can be done in parallel