UNIVERSITY of HOUSTON

Homework #1

Due 11:59pm, Monday, February 3, 2020
Multiple submissions accepted, last one graded.
100 points total.

Name: Xena Toumajian

PeopleSoft ID: 1662518

When you see something like <COD §1.7>, it is referring you to a section in the textbook: COD means Computer Organization and Design, and § 1.7 means section 1.7.

1.2 (16 pts) <COD §1.2> The eight great ideas in computer architecture are similar to ideas from other fields. Match these eight ideas from computer architecture:

- **a.** Design for Moore's Law
- **b.** Use Abstraction to Simplify Design
- **c.** Make the Common Case Fast
- **d.** Performance via Parallelism
- **e.** Performance via Pipelining
- **f.** Performance via Prediction
- **g.** Hierarchy of Memories
- **h.** Dependability via Redundancy

to the following ideas from other fields. Write your answer in the first column:

e	Kitchen cooktops usually have 4 or 6 burners. Some people have 2 ovens or 2 microwaves.
c	Stores like WalMart have lanes for people with 20 items or less because a large percentage of their customers buy less than that number of items.
d	https://www.youtube.com/watch?v=HnbNcQlzV-4
g	Ford is building marketing material to encourage people to take long 500-mile vacation trips in their electric SUV even though today nothing exists that can go that far.
b	The city of Brasília, Brazil, invented in 1960 as its new capital, was designed largely by Oscar Niemeyer in the shape of an airplane. The details of building the streets was left for others.

h If I don't find the book I want on the shelf in my classroom, I can always go to the library.

a Most cars have spare tires.

f Nurses in operating rooms keep the instruments they feel the surgeon is most likely to request on the tray closest to the surgeon..

1.4 (16 pts) <COD §1.4 > Assume the built-in LCD color display on your laptop activates each LCD pixel on the display according to data it finds in a special buffer (memory) that tells it what each pixel should be doing, thus producing the image you see on the screen. If will use 8 bits for each of the red, green and blue pixels and our laptop has a display of size 1900 x 1200 pixels (which we call a frame), then answer the following:

a. (4 pts) What is the minimum size in <u>bytes</u> of the frame buffer to store a single one of these screen images?

Total pixels= 1900x1200= 2280000
Total bytes= 2280000*3= 6840000

b. (4 pts) While one frame (image) is being displayed to the user by the monitor, you have just a moment to build the next frame. As mentioned before, it is built and stored in a frame buffer, a special location in memory. Once built, you have to transmit it to the display over a data bus (a type of network) as digital data. If the bus feeding the data from this frame buffer to your monitor operates at 500 Megabits per second (Mbps), how long will it take to send one frame? Answer to the hundredth of a second. *Hint: assume a megabit is 10^6 bits.*

6840000x8/(500x10^6)=0.10944

c. (4 pts) How many entire (not fractional) frames per second can this bus transmit?

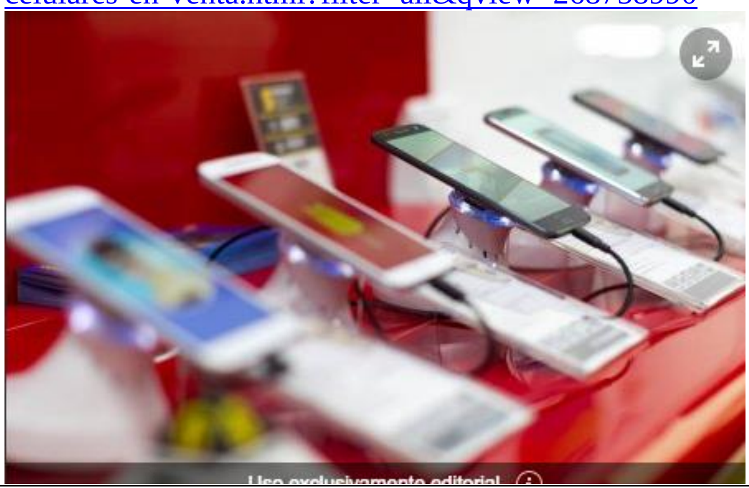
1/0.10944= 9.137   9

d. (4 pts) Rounded to the nearest megabit per second, if you want the system to refresh the screen 60 times per second, what bus speed must you have?

3.2832

P3. (16 pts) Which of the following attributes apply to CISC and RISC architectures? You may refer to the Stanford article from lecture 2 (not part of Zybook). Mark one or both or neither.

CISC	RISC	Attribute
☒	☐	The same C program is compiled on two different computers, one ARM and one CISC. On which will the smaller assembly program be found?
☒	☐	On which system will more CPU cycles be needed to run those compiled programs?
☐	☒	Which will likely require more time to compile?
☒	☐	Which is better suited for mobile devices with limited memory?

■	☐	What everything in this photo is based on: https://mx.depositphotos.com/73918051/stock-photo-shelves-with-laptops-in-store.html 
☐	■	What these are almost certainly all based on: https://mx.depositphotos.com/stock-photos/telefonos-celulares-en-venta.html?filter=all&qview=268758990 
■	☐	Exemplified by Intel's x86 architecture
☐	☐	Currently ARM is a prominent example of approach.

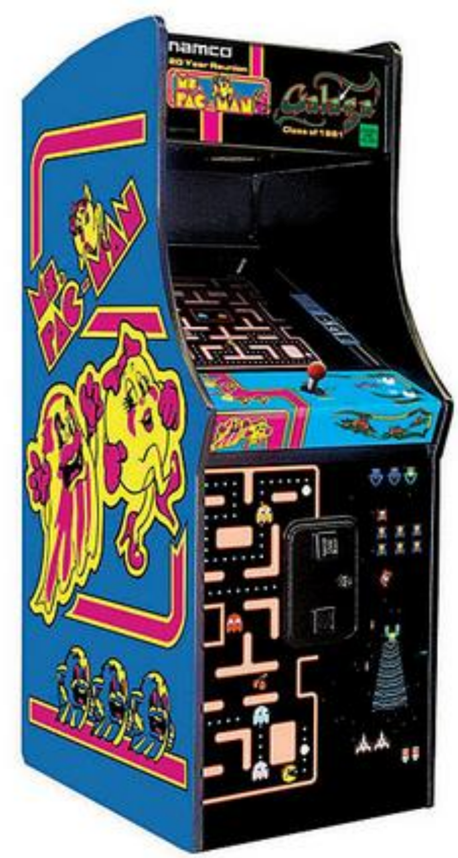
P4. (14 pts) We discussed Intel's x86 architecture and its recent rival, ARM. ARM is used in embedded applications, Intel in just about everything else. From our alternate textbook by Stallings (refer to the syllabus), we are taught that "*The term embedded system refers to the use of electronics and software within a product, as opposed to a general-purpose computer, such as a laptop or desktop system.*"

Embedded doesn't mean that the product can't have different programs that run, but at a system level it's built-in to a specific hardware device and the OS is part of the system and unlike a laptop, cannot be changed, and is usually developed by the embedded system provider.

A *deeply-embedded system* is a subset of embedded, and is defined as a system dedicated to a single purpose, whose software and architecture are typically obscured from the user. Here there is no way to add more programs, or even see how to do that; it's a black box, as we say – the inner-workings and details of which are not visible to us.

Can you categorize into which type of system each of the examples falls? Think about whether you can load another program and add to what the system is doing. If so, then it's not *deeply* embedded. Could you write your own code to run directly on the chip like a custom OS, replacing what the manufacturer had in mind, without destroying its original purpose? If so, it's probably not an embedded system either.

GP = General Purpose, EM = Embedded System and DE = Deeply Embedded.

GP	EM	DE	Attribute
☑	☐	☐	An MP3 music player, like an Apple iPod.
☐	☑	☐	Your typical Television remote control
☑	☐	☐	A Smart TV with an app storel
☑	☐	☐	An Apple or Samsung tablet computer
☐	☐	☑	A pacemaker to regulate heartbeats
☐	☑	☐	A Nest thermostat
☐	☐	☑	This device as originally built: 

☐	☒	☐	If you changed the video game so that you could download new games from a USB thumb drive.
☐	☐	☒	The raspberry Pi as described here: https://en.wikipedia.org/wiki/Raspberry_Pi#Overclocking
☐	☐	☒	This raspberry Pi-based device: https://www.trafficsafetywarehouse.com/pi-Lit-Sequential-Barricade-Style-Lamp-Case-of-10/productinfo/BASEQ-YE1/
☐	☐	☒	The system that regulates the mix of gasoline and oxygen in your engine to ensure efficient use of fuel.
☐	☐	☒	This device: https://www.sparkfun.com/products/9963?_ga=2.223621211.1155801348.1580329393-796445367.1580329393
☒	☐	☐	Your desktop or laptop
☐	☒	☐	An open-source router that lets you choose from multiple types of firmware (an OS stored in HW instead of on a drive) like the linksys device described here: https://www.flashrouters.com/learn/router-basics/benefits-of-open-source-firmware

1.5 (24 pts) <COD §1.6> Consider 2 different processors P1, P2 executing the same instruction set.
P1 has a 4.18 GHz clock rate and a CPI of 2.07.
P2 has a 2.52 GHz clock rate and a CPI of 5.14.

a. (4 pts) What's the performance of each processor expressed in instructions per second (IPS)? *Show the equations you used.* Clock Rate/CPI

P1: 4.18x10^9/2.07= 2.019x10^9

P2: 2.52x10^9/5.14 = 4.9x10^9

Which processor has the highest performance? ☐P1 ☒P2

b. (4 pts) If the processors each execute a program in 15 seconds, find the number of cycles for each processor. Your answer should be in scientific notation, one non-zero digit to the left of the decimal, let's say 2 digits to the right, and multiplied by a power of 10. For example, 1.23×10^9.

P1: 3.02x10^10 cycles

P2: 7.35x10^10 cycles

c. (4 pts) And now find the number of instructions executed in this time for each processor. *Make sure you use enough digits of significance to uncover any* *difference; don't let rounding make it look like they have run the same number if in fact they have not.*

P1: 6.27x10^10 cycles P2: 3.78x10^10 cycles

d. (4 pts) We discover we can reduce the execution time for each processor by 20% but it leads to an increase of 10% in the CPI. What is the new execution time for both processors? Always show your formulas.

New execution time: 12 sec

e. (4 pts) What is the new CPI for each processor?

P1 new CPI: 2.27 P2 new CPI: 5.654

f. (4 pts) We want to reduce the original execution time of each CPU by 20%. But this time the CPIs remain unchanged. Recalculate the clock rate (cycles/sec or GHz) that will be required to achieve this time reduction. We calculated the total instructions required to be processed above; that won't change. What's the new clock rate in a max of 4 digits of significance?

P1: 10.8 GHz P2: 16.191 GHz

(Probably way too much) Background:
Keep in mind:

- The clock rate is the inverse of the interval of a single clock pulse.
- The pulse duration is referred to as a *clock cycle*.
- Clock rate is expressed in hertz (often gigahertz or GHz or megahertz or MHz)
- Clock cycles are expressed in nanoseconds (ns) or picoseconds (ps).

In your solution, convert to the appropriate metric unit prefix. This is not exactly the same thing as scientific notation, because there is not a named unit for every power of 10; rather for every three powers of ten (or powers of 1000). Therefore you may have a 1-, 2- or 3-digit decimal number of significance (greater than 0) to the left of the decimal point, but not a number ≤ 1 nor a number ≥ 999 left of the decimal point. In those cases, we are using the wrong prefix. You may also not mix and match any exponent notation with these prefixes; do that with the base unit.

Correct Examples	
1.928 gigahertz	1.928 picoseconds
92.88 GHz	144.93 nanoseconds
800.12 GHz	-299.3 ps
999.99 gigahertz	140 milliseconds
140 GHz	2.03 ms

4.37×10^{-8} Hz	4.37×10^{15} seconds

Incorrect Examples		Corrected Examples	Explanation
4.37×10^{2} gigahertz	→	437 gigahertz	Mixing exponents and prefixes
1,203.0 gigahertz	→	1.2030 terahertz	≥1000, need to move up to next prefix
0.488 gigahertz	→	488 megahertz	≤1, move down
4.37×10^{2} ms	→	437 ms or 4.37×10^{-1} s	Same rules for seconds

Here's a table from https://en.wikipedia.org/wiki/Metric_prefix:

SI prefixes					
Prefix		Base 1000	Base 10	Decimal	English word
Name	Symbol				Short scale
yotta	Y	1000^{8}	10^{24}	1000000000000000000000000	septillion
zetta	Z	1000^{7}	10^{21}	1000000000000000000000	sextillion
exa	E	1000^{6}	10^{18}	1000000000000000000	quintillion
peta	P	1000^{5}	10^{15}	1000000000000000	quadrillion
tera	T	1000^{4}	10^{12}	1000000000000	trillion
giga	G	1000^{3}	10^{9}	1000000000	billion
mega	M	1000^{2}	10^{6}	1000000	million
kilo	k	1000^{1}	10^{3}	1000	thousand
hecto	h	$1000^{2/3}$	10^{2}	100	hundred
deca	da	$1000^{1/3}$	10^{1}	10	ten
		1000^{0}	10^{0}	1	one
deci	d	$1000^{-1/3}$	10^{-1}	0.1	tenth
centi	c	$1000^{-2/3}$	10^{-2}	0.01	hundredth
milli	m	1000^{-1}	10^{-3}	0.001	thousandth
micro	μ	1000^{-2}	10^{-6}	0.000001	millionth
nano	n	1000^{-3}	10^{-9}	0.000000001	billionth
pico	p	1000^{-4}	10^{-12}	0.000000000001	trillionth
femto	f	1000^{-5}	10^{-15}	0.000000000000001	quadrillionth
atto	a	1000^{-6}	10^{-18}	0.000000000000000001	quintillionth
zepto	z	1000^{-7}	10^{-21}	0.000000000000000000001	sextillionth
yocto	y	1000^{-8}	10^{-24}	0.000000000000000000000001	septillionth

A simpler table can be found in table 3 here: https://www.btb.termiumplus.gc.ca/tcdnstyl-chap?lang=eng&lettr=chapsect1&info0=1.23

1.6 (16 pts) <COD §1.6> Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI . We'll label them A: Integer, B: Floating Point, C: Load/Store and D: Control. Assume our program runs for 2 million instructions.

More data you'll need:

P1 Clock rate	5.92E+09	Hz
P2 Clock rate	3.77E+09	Hz

	P1 CPI	P2 CPI	IC %
Class A	4.39	4.61	9.00
Class B	8.85	10.44	4.00
Class C	14.04	8.72	69.60
Class D	12.56	11.86	17.40

To walk through an example, the Integer instructions, which we labeled A, require 4.39 cycles per instruction on P1, 4.61 on P2, and represent 9% of all instructions, or 180,000 of the 2M instructions.

a. (8 pts) How many instructions are run in each category?

Class A	180000
Class B	80000
Class C	1392000
Class D	348000
Total	2.00E+06

b. (4 pts) What is the global CPI for each implementation?

For P1: 12.706 For P2: 8.965

c. (4 pts) Calculate how long each CPU will spend calculating all of the cycles required to run the program's instructions, given the clock speed of each CPU. Answer in the proper prefix to seconds as described in 1.5(f), not in scientific notation. Use 4 digits of significance.

P1: 0.1689x10^-9 P2: 0.2652x10^-9