Introduction to Computer Systems and Platform Technologies

Study Period 3, 2021 – CPT160

Assessment 1

Student Name:	Adam Mutimer
Student Number:	S3875753

Contents

Task 1 – Number Systems 3

Task 2 – Binary Addition and Subtraction 6

Task 3 – Bitwise Operations 8

Task 4 – Logic Circuits and Truth Tables 9

Task 5 – Pipelining 10

Task 6 – CPU Architecture 11

Task 7 – Memory 14

Task 8 – Hamming & SECDED Code 15

Advanced Questions 17

References / Bibliography 18

- Task 6 – Question A 18
- Task 6 – Question B 18
- Task 6 – Question C 18
- Task 7 – Question A 18
- Task 7 – Question B 18
- Task 7 - Question C 18

Task 1 – Number Systems

Student Number: s3875753
X = 5753

a) Convert X from decimal to binary

Division by 2	Quotient	Remainder	Bit#
5753/2	2876	1	0
2876/2	1438	0	1
1438/2	719	0	2
719/2	359	1	3
359/2	179	1	4
179/2	89	1	5
89/2	44	1	6
44/2	22	0	7
22/2	11	0	8
11/2	5	1	9
5/2	2	1	10
2/2	1	0	11
1/2	0	1	12

X = 0001 0110 0111 1001

b) Convert the binary string obtained from your answer to (a) into an octal and hexadecimal

a. Octal Conversion

Octal Digit Value	Binary Equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

001	011	001	111	001
1	3	1	7	1

Octal = 13171

b. Hexadecimal Conversion

Hex Value	Binary Equivalent
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101

0001	0110	0111	1001
1	6	7	9

Hexadecimal = 1679

c) Convert X from decimal to base 14, Where A, B,C and D correspond to 10, 11, 12 and 13 respectively.

$$\begin{array}{r} 410 \\ 14\overline{)5753} \end{array} = 14 \times 410 = 5740 = \frac{5753}{5740} = \underline{13}$$

$$\begin{array}{r} 29 \\ 14\overline{)410} \end{array} = 14 \times 29 = 406 = 410 - 406 = \underline{4}$$

$$\begin{array}{r} 2 \\ 14\overline{)29} \end{array} = 14 \times 2 = 28 = 29 - 28 = \underline{1}$$

$$\begin{array}{r} 0 \\ 14\overline{)2} \end{array} = 14 - 2 = 12 = 14 - 12 = \underline{2}$$

$$\underline{214D}$$

d) **Now add 42^{10}(42 in decimal) to X and calculate the sum in base 14. Consider the following two calculations:**

   i. **Conversion (base 14 to decimal) before addition (in base 14): convert 42^{10} in to base 14, then add the two base 14 numbers.**

$$\begin{array}{l} \quad\ \ 3 \\ 14\overline{)\ 42} \end{array} = \underline{0}$$

$$14\overline{)\ 3} = 14-3 = 11 = 14-11 = \underline{3}$$

$$30_{14}$$

$$\begin{array}{r} 214D \\ +\quad 30 \\ \hline \underline{217D} \end{array}$$

   ii. **Addition (in decimal) before conversion (decimal to base 14): add 42^{10} to X in decimal, then convert the decimal sum into base 14.**

$$5753 + 42 = 5795$$

$$\begin{array}{l} \quad\ \ 413 \\ 14\overline{)\ 5795} \end{array} = 14 \times 413 = 5782 = 5795 - 5782 = \underline{13}$$

$$\begin{array}{l} \quad\ \ 29 \\ 14\overline{)\ 413} \end{array} = 14 \times 29 = 406 = 413 - 406 = \underline{7}$$

$$\begin{array}{l} \quad\ \ 2 \\ 14\overline{)\ 29} \end{array} = 14 \times 2 = 28 = 29 - 28 = \underline{1}$$

$$14\overline{)\ 2} = 14 - 2 = 12 = 14 - 12 = \underline{2}$$

$$\underline{217D}$$

   iii. **Which calculation is simpler? Please explain you answer. How many digits are different from your answer to (c)?**

In this instance I found the conversion method simpler, however I do prefer the addition before conversion as its easy to follow when going back to review the work as it makes more sense to me personally. Only 1 digit was different from my answer to (c) 214D is now 217D, 4 being the single digit that has changed to a 7.

e) **Consider a base 26 number system wherein the letters of the alphabet are the digits.**
That is, A=0, B=1, C=2, ….Z=25 in base 10.
Use the first three letters of your given name as a number in the base 26 system, and the first three letters of your surname as another number in the base 26 system.

Add these numbers together to obtain the sum in base 26.

ADA = A=0, D=3, A=0
MUT = M=12, U=20, T=19

Base 26	A	D	A
➔ Base10	0	3	0
Base 26	M	U	T
➔ Base10	12	20	19
+ (Addition Base10)	12	23	19
= (SUM Base26)	M	X	T

SUM = MXT

Task 2 – Binary Addition and Subtraction

a) **Convert the decimal numbers A and B to 4-bit binary numbers. (A=3 & B=5). Show hoe to add together these two 4-bit binary numbers and state whether the answer is valid to 4-bit arithmetic.**

Convert A (3^{10}) to Binary			
Division by 2	Quotient	Remainder	Bit#
3/2	1	1	0
1/2	0	1	1
0/2	0	0	2

A = 011 → 4-Bit = 0011^2

Convert A (5^{10}) to Binary			
Division by 2	Quotient	Remainder	Bit#
5/2	2	1	0
2/2	1	0	1
1/2	0	1	2

B = 101 → 4-Bit = 0101^2

Carry	1	1	1	
	0	0	1	1
+	0	1	0	1
=	1	0	0	0

SUM = 1000^2 this is answer is valid to 4-bit arithmetic

b) Convert the decimal numbers A and B to 5-bit binary numbers. Using two's compliment representation, show how to:

A = 0011 B= 0101

16's	8's	4's	2's	1's	TASK
	0	*0*	*1*	*1*	*4-bit*
0 (Positive Number)	*0*	*0*	*1*	*1*	*5-Bit*
0	*1*	*1*	*0*	*0*	*1's Compl.*
1 (Flipped this to become a negative)	*1*	*1*	*0*	*1*	*2's Compl.*

To confirm 11101 = -3:

-16 + 8 + 4 +1 = -3

i. Subtract the two 5-bit binary numbers (-A-B)

We need to make (A) a negative number, so we flip bits and add 1 = A=11101

$\mathbf{11101^2 - 00101^2}$

$2^{n1} = 2^{5^{-1}} = 2^4 = 16$

	16's	8's	4's	2's	1's
CARRY	1	1		1	
	0	0	1	0	1
+	1	1	1	0	1
1 (Overflow Ignore)	0	0	0	1	0

***00010 = 2* to confirm this:**

0 (16's) + 0 (8's) + 0 (4's) + 1 (2's) + 0 (1's) = 2

To confirm:

As a decimal A= -3 and B = 5

-3 + 5 (5 -3) = 2

SUM = 00010

ii. How to translate the binary result back to decimal.

(Note if your solution is negative, you must use 2's compliment to show the positive equivalent)

#4	#3	#2	#1	#0
0	0	0	1	0
$\mathbf{0\ x\ (2^4)}$	$\mathbf{0\ x\ (2^3)}$	$\mathbf{0\ x\ (2^2)}$	$\mathbf{1\ x\ (2^1)}$	$\mathbf{0\ x\ (2^0)}$
0	0	0	2	0

0 + 0 + 0 + 2 + 0 = 2

SUM = 2

Task 3 – Bitwise Operations

A AND M = $\mathbf{1111\ 0000^2} \boldsymbol{AND}\ \mathbf{0111\ 1111^2 = 0111\ 0000^2}$

AND Truth Table		
Input	Mask	Result
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table		
Input	Mask	Result
0	0	0
0	1	1
1	0	1
1	1	1

XOR Truth Table		
Input	Mask	Result
0	0	0
0	1	1
1	0	1
1	1	0

NOT AND Truth Table		
Input	Mask	Result
0	0	1
0	1	1
1	0	1
1	1	0

a) Reset but 0, bit 7 and leave the reset untouched

Bit Position	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
Input	0	1	1	1	0	0	0	0
Mask	0	1	1	1	1	1	1	0
Result of AND **Operation**	0	1	1	1	0	0	0	0

Result: $0111\ 0000^2$

b) Make sure that bit 2 and bit 6, and only these, are reset, the others are set
My understanding of this question is that bit 2 and 6 need to be reset to 0 while all other bits are set to 1

Bit Position	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
Input	0	1	1	1	0	0	0	0
Mask	1	1	0	0	1	0	1	1
Result of XOR Operation	1	0	1	1	1	0	1	1

Result: $1011\ 1011^2$

c) Toggle the values of the middle 4 bits (the opposite of what thewy are currently) and set the 2bits on each side.

Bit Position	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
Input	1	0	1	1	1	0	1	1
Mask	0	1	1	1	1	1	0	0
Result of NAND **Operation**	1	1	0	0	0	1	1	1

Result: $1100\ 0111^2$

Task 4 – Logic Circuits and Truth Tables

a) **Write down the equivalent logic expression (simplification is NOT required)**
Circuit 1: $!A\ AND\ B\ OR\ !C$
Circuit 2: $!A\ AND\ B\ OR\ !C$

b) **Write a truth table that shows the final output (O) for inputs A, B and C (Showing all your working out and intermediate steps, i.e., the output of each gate, in the truth table is a column)**
Note: NOT gates flip the result of the source

Circuit 1:

INPUTS			Gate Output					Outputs
A	B	C	1	2	3	4	5	O
0	0	0	0	0	1	1	1	TRUE
1	0	0	0	0	1	1	1	TRUE
1	1	0	1	0	1	1	1	TRUE
1	1	1	1	1	1	0	0	FALSE
0	1	0	0	0	1	1	1	TRUE
0	1	1	0	0	1	1	1	TRUE
0	0	1	0	0	0	1	0	FALSE
1	0	1	0	1	0	1	0	FALSE

Circuit 2:

INPUTS			Gate Output		Outputs
A	B	C	6	7	O
0	0	0	0	1	TRUE
0	1	0	1	1	TRUE
0	0	1	0	0	FALSE
0	1	1	1	1	TRUE
1	0	0	0	1	TRUE
1	1	0	0	1	TRUE
1	1	1	0	0	FALSE
1	0	1	0	0	FALSE

c) **Compare the final output columns in the two truth tables. Do these two expressions give the same output?**
Hence, are the 2 expressions equivalent?

Yes, the two expressions return the same output, circuit one being a more over engineered method while circuit two being a more simple and elegant method. Hence the two expressions are the same

Task 5 – Pipelining

Task	**Sequential – Carrier Pigeon**								
Writing	🟥			🟨			🟦		
Fanning		🟥			🟨			🟦	
Catching			🟥			🟨			🟦
Time	25 Min (			10 Min			15 Min		

Task	**Pipelined – Carrier Pigeon**								
Writing	🟥	🟨	🟦						
Fanning		🟥	🟨	🟦					
Catching			🟥	🟨	🟦				

a) How long does it take for all to send messages sequentially?
How long does it take for all of them to send messages pipelined?

In this example it will take both methods 3.33hours to send all the messages, however the only difference between the two methods is that using a pipelined method all four messages' tasks are grouped together and processed in batches (4x writing, 4x fanning, 4x catching) which they will still take 25min each, 10min each or 15min each; Messages will be sent to the destination at roughly at the same time 15min apart.

However, using the sequential method which involves the handling of only a single message at a time and its tasks as a single job (writing, fanning, catching) messages will be sent roughly 50min apart which is the time it takes to complete the 3 individual tasks for a single message.

b) How and why does pipelining help with the throughput of entire workloads?
The workloads involved sending 4 messages as described above, in comparison to sending 4 messages sequentially.

Consider Robin, Bryan, Finchie and Dan as processes waiting for individual workloads (messages) to process and those workloads(messages) are constantly generated; in the pipeline method all four processes would have workloads to process 15min +/- apart every 50min and for the majority of the time not be idle.

In the sequential method. Robin would be busy after 50min; however, Bryan will be idle for a further 50min after Bryan has a workload, and Finchie will be waiting 1.66hours for a workload from when Robin receives a workload, Dan will be waiting the longest at 2.5hours for its first workload.

Taking into account this laymen's explanation of the two pipelines would be able to process more workloads consistently (throughput) while maintaining a low idle process time. Whereas sequentially processing results in much higher idle time of the processes (Robin, Bryan, Finchie and Dan)

c) **Can pipelining help reduce the latency of any one step in the 4 people sending a message in the scenario as described above?**
Show how the pipeline rate is limited by the slowest pipeline stage

In this scenario pipelining would only reduce the time between each message being sent; for example, all messages will still take 3.33hours to process as a whole. However, the lag time between each message being sent will be reduced as they will be sent roughly 15min apart as the 3 individual tasks for each message are processed in group batches.

Task 6 – CPU Architecture

a) **Compare and contrast "multithreading" and multiprocessing" in terms of hardware**
The most obvious difference between multi-processing and multi-threading is that multi-processing requires more than 1 CPU, whereas multi-threading only requires a single CPU.

My understanding from reading the three sources mentioned in the ***References / Bibliography*** is that multi-threading is a programming execution technique designed to allow a single process to be split into multiple segments that a single CPU and run simultaneously/parallel. Multiprocessing involves multiple CPUs to increase the computational power that a task/process has available and from my understanding in theory can also make use multi-threading on the multiple CPU's

Based on the 3 sources that I found most useful I have created a comparison table outlining differences between the two methods:

	Multi-Processing	**Multi-Threading**
Supports Multiple CPU's	YES	NO
Supports Multiple CPU Cores	YES	NO
Individual Process can be killed	YES	NO
Rapid Process Creation Time (Time it takes to prepare a workload)	NO	YES
Independent Memory Address Space (Each Process or Thread has its own memory allocation)	YES	NO
Threads or Processes are lightweight (Memory Allocation)	NO	YES
Rapid Task-Switching	NO	YES

Even though multi-processing has its advantages its not ideal for a lot of tasks on its own; tasks that would benefit from multi-processing would be Video Rendering, Scientific Calculations, that require large amounts of computational power. Now we need to keep in mind that current generation CPU's are no longer just a single CPU; they are multi-core CPU's, each core is a CPU in its own right however they do share cache memory and a few other component's. therefor a program written to use the multi-processing technique would be able to utilize all cores of a current generation CPU of the work load was great enough and in a multi-socket system like AMD EPYC would be able to use two CPUs or more and all cores of those CPU's.

Multi-threading is perfect for tasks which require less computational power but has tasks that need to running in parallel to each other to keep latency down; such as a word processor or video game; for an example I'm going to use Elder Scrolls Online as an example; this game is broken down into multiple threads running parallel to each other, one thread is maintain the connection to the upstream server, one is taking care of the U.I (user interface) whilst another is sending and receiving data to and from your graphics card for generating the in game environment to display on your monitor.

Traditionally such games in previous years have been really monolithic thread orientated but to current gen gaming consoles a shift in the industry is promoting multi-thread optimized games to take advantage of the console hardware and the graphics units that power them.

An example of a Monolithic thread game would be a M.U.D (multi-user dungeon) a multiplayer text-based game from the early 90's and late 80's. in particular MudOS based LPMuds had a single process/thread and could not take advantage of multi-processing or multi-threading. Which limited the computational power that was available to perform in game tasks.

b) Explain how threads are used by the CPU to process tasks by describing a modern example, e.g., the multi-core mobile phone that you use every day has an interesting organization of threads.
However, it can be any other modern example of hardware that uses "threads".

Threads are processes that can split into multiple sub processes to be processed in parallel, Intel CPUs call this process hyper-threading, AMD & ARM call this simultaneous multithreading or SMT for short both have the same outcome but the manner in which it is achieved is different due to patients.

Using the example mentioned in the question "multi-core mobile phone" which in my case is an android phone; Samsung Galaxy S21 which is using a **Snapdragon 888 SOC** which contains a CPU based on Arm Cortex-X1 design.

The Arm Cortex-X1 is a 64bit Advanced Risk Machine CPU with 8 CPU cores in total 4x cores running at 2.3Ghz and 4x cores running at 1.6Ghz.
This processor uses SMT (Simultaneous Multithreading similar to that of AMD).

SMT is a process in which a CPU is able to split a physical core and create an extra virtual core allowing it to run two processes simultaneously on that core. Applications with processes that are designed for multi-threading are able to take full advantage of this in order to increase computational power and reduce latency of a task. When an application is written to take advantage of this the CPU is able to split the process in to multiple smaller tasks/threads and process them in parallel rather than sequentially.

c) What is a warp GPU architecture and what is the major constraint of its operation?

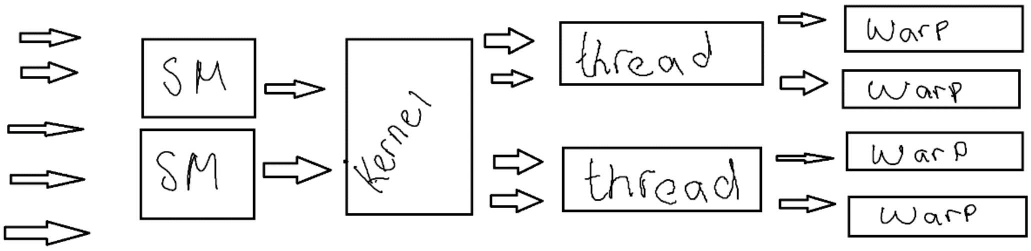
Firstly, GPUs and CPU while similar in many aspects use different terms, the table bellow explains this:

GPU Term	Quick Definition for a GPU	CPU Equivalent
Thread	The stream of instructions and data that is assigned to one CUDA core; note, a Single Instruction applies to Multiple Threads, acting on multiple data (SIMT)	N/A
CUDA Core	Unit that processes one data item after another, to execute its portion of a SIMT instruction stream	Vector Lane
Warp	Group of 32 threads that executes the same stream of instructions together, on different data	Vector
Kernel	Function that runs on the device; a kernel may be subdivided into *thread blocks*	**Threads(s)**
SM, streaming multiprocessor	Unit capable of executing a thread block of a kernel; multiple SMs may work together on a kernel	**Core**

Source: https://cvw.cac.cornell.edu/GPUarch/threadcore

A Warp is a group of threads currently 32 that will execute the same stream of instructions together on different dataset simultaneously. Multiple Warps can be processed simultaneously by a single SM (Streaming multiprocessor)

To better explain this, I have created a diagram of my understanding of the process flow:



A major constraint of its operation is performance penalties when two or more parts of the same "warp" branch to different instruction sets, complicating the execution flow of the warp; I could not find further information on this constraint to elaborate further.

Task 7 – Memory

a) **Describe the main features of INTEL’s “Optane Memory”?**

Feature	Description
< 1 Microsecond Latency	This feature is claiming access times to the memory from the CPU are less than 1 Microsecond, 1 Microsecond is 1 second divided by 1,000,000 and is close to native CPU cache speeds
Memory has three modes	**App Direct Mode:** Data in memory is retained even after suffering a power loss or restart (Persistent Memory) **Memory Mode:** Enables applications to use “Intel Optane PMem (Persistent Memory) as expanded volatile system memory. **Storage Over App Direct Mode:** Allows memory to be used as a persistent storage device similar to an SSD
High Capacity	Module Sizes up 512GB

b) **What are some of the claims made by the “Intel Marketing Department” regarding to what Optane Memory provides?**

- Less than 1 Microsecond latency
  - This would be in the most ideal and optimized environment.
- No other memory that exists today can customize your system’s performance like Intel® Optane™ memory
  - Some of features claimed to be unique are just enhancements on already existing technology such as RAM disks (Using Memory as Storage), Persistent memory storage devices have previously been available with independent backup power to keep data persistent during power outages and reboots
- Close the gap between traditional memory and storage with Intel® Optane™ memory, a standalone memory device that is different from all the rest. It streamlines your computing experience by combining storage capacity and intelligent system acceleration. You get both speed and capacity—all at a low price.
  - This is just a marketing gimmick, designed to attract gamers, etc

DDR5 memory is the latest computer memory being developed and will eventually replace DDR4 memory

c) **What are the maximum clocks speeds for DDR4 memory, and what is proposed for DDR5?**

DDR4’s Maximum clock speed is 3200Mhz and DDR5 proposed maximum is 8400Mhz

d) **What is the current maximum throughput available for DDR4 and what is proposed for DDR5?**

DDR4’s Maximum throughput at 3200Mhz is 25.6 GB/s
DDR5’s Maximum proposed throughput is up to 90 GB/s

Task 8 – Hamming & SECDED Code

a) **For data, using 4 Hamming code parity bits determine the maximum number of data bits that can be protected**

BIT #		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoded data Bits		P1	P2	D1	P4	D2	D3	D4	P8	D5	D6	D7	D8	D9	D10	D11
Parity Coverage	P1	X		X		X		X		X		X		X		X
	P2		X	X			X	X			X	X			X	X
	P4				X	X	X	X					X	X	X	X
	P8								X	X	X	X	X	X	X	X

Using 4 hamming code parity bits the total maximum length will be 15 – 4 parity bits leaving a total of 11 data bits that can be protected.

b) **A SECDED encoded character has been retrieved, with the hexadecimal value $E76^{16}$. You may assume that the SECDED parity is even.**

Convert Hexadecimal to binary:
$E76^{16}$ = (E = 1110) (7= 0111) (6=0110) = $1110\ 0111\ 0110^{2}$

11	10	9	8	7	6	5	4	3	2	1	0	BIT #	
D7	D6	D5	P8	D4	D3	D2	P4	D1	P2	P1	P0		
1	1	1	0	0	1	1	1	0	1	1	0	**DATA**	
											X	**P0**	
1		1		0		1		0		1		**P1**	
1	1			0	1			0	1			**P2**	
				0	1	1	1					**P4**	
1	1	1	0									**P8**	

Received:
P0 = 0 (SECDED BIT)
P1 = 1
P2 = 1
P4 = 1
P8 = 0

Calculated: (Assuming even parity)
P1 = XOR(0,1,0,1,1) = XOR (0,1,0,1,1,0) = 1
P2 = XOR(0,1,0,1,1) = XOR (0,1,0,1,1,0) = 1
P4 = XOR(1,1,0) = XOR (1,1,0,0) = 0
P8 = XOR(1,1,1) = XOR(1,1,1,1) = 0

P0 = (Count 1's received exclude bit# 0/p0) = 8x 1's => 0

1. **Was there an error in transmission? Explain your answer.**
   Yes, there was an error detected. Parity check bits received did not match the parity bits calculated.
   Parity Bits received: 0111 0
   Parity Bits Calculated: 0011 0

   According to the Single Error Detection bits we have an error in P4, however P0 is indicating no error.

2. **If there was an error, either correct it (Reporting the corrected ASCII character) or explain why it could not be corrected (Show your Hamming/SECDED table)**

P8	P4	P2	P1	P0	
0	1	1	1	0	Received
0	0	1	1	0	Calculated
0	1	0	0	0	Syndrome

$\underbrace{\quad 0 \quad 1 \quad 0 \quad 0 \quad}_{\text{SEC}} \underbrace{\quad 0 \quad}_{\text{DED}}$

$0100_2 = 4_{10} = \underline{\underline{P4}}$ has been flipped

Based on my calculations with the assumption of even parity, only parity bit 4 has been flipped indicating our data is intact. However, for completeness' sake the actual binary message should have been: $1110\ 0110\ 0110^2$ and decoded is: 110110^2

The ASCII representation is: **v**

Advanced Questions

References / Bibliography

Task 6 – Question A

https://www.geeksforgeeks.org/difference-between-multiprocessing-and-multithreading/

https://www.guru99.com/difference-between-multiprocessing-and-multithreading.html

https://techdifferences.com/difference-between-multiprocessing-and-multithreading.html

Task 6 – Question B

https://www.qualcomm.com/products/snapdragon-888-5g-mobile-platform

https://www.tomshardware.com/reviews/simultaneous-multithreading-definition,5762.html

https://developer.arm.com/documentation/#sort=relevancy&f:@navigationhierarchiesproducts=[IP%20Products,Processors,Cortex-X]

Task 6 – Question C

https://nyu-cds.github.io/python-gpu/02-cuda/

https://cvw.cac.cornell.edu/GPUarch/simt_warp

https://cvw.cac.cornell.edu/GPUarch/threadcore

Task 7 – Question A

https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/what-is-optane-technology-brief.pdf

Task 7 – Question B

https://news.skhynix.com/why-ddr5-is-the-industrys-powerful-next-gen-memory/

https://www.transcend-info.com/Support/FAQ-292

Task 7 - Question C

https://www.notebookcheck.net/Leaked-DDR5-6400-RAM-test-on-Intel-s-upcoming-Alder-Lake-platform-shows-that-faster-DDR5-specs-are-not-really-suitable-for-gaming.561114.0.html