EEE4120F 2023
Class Test 1
including Comprehension Test Attempt 2

Date: 5 April 2023

Time: 14:00-15:30 *Duration:* 1h 30min 100 marks

Venue: Menzies EM6

SOLUTIONS

Procedures:
please read this first page while waiting for the test to start!

Answer all questions (including all GA questions)

NB! ***If using back of page for answer make sure to indicate the question number!***

This test is planned to be venue-based. This is a closed-book assessment, no use of shartphones or other electronics gadgets permitted, although you may use a calculator. The requisite procedures must be followed at all times.

NOTE: ANSWER ON THIS QUESTION PAPER IN SPACES PROVIDED AFTER THE QUESTIONS.
Please Make sure your name and student number in clearly indicated above. Any additional loose papers you might use, please ensure these each have your student number and "GA2 15-03-2023" indicated (these should be stapled to your other answers).

If finished early: If completed early, and before the last 10 minutes of the test, please signal to the invigilator that you are done, so that you your paper is collected and then you can quietly leave.

When finished: Check that your **name and student number is clearly filled in on this page** and also on any other pages you might have used and want included with your answers.

NOTE: Questions Q1 – Q5 are all GA related. Note about GA pass/fail: even if you have passed the GA questions in the previous test, do please still answer these or marks will be lost. You need pass a GA question only once to have that aspect recorded as a pass (so if you passed a GA question last test and fail it in this test you would still pass the GA but not get a pass mark for that question in the test).

Recap of conceptual project

The Multi-point Pulsar Identifying and Correlation Cluster (MPICC)

A basic block diagram of the MPICC is given below. As the illustration shows, there are multiple types of nodes in the system. Some of which have ADCs, and other processing needs applied to sampled signals to generate images, that are then collected and sent to an intermediate 'MPICC hub'. The hubs have some computing resources, which are applied to do further cleaning of the data and to do partial processing and filtering.

Something to bear in mind is that parts of this system are planned around being more shoestring budget than other parts. Overall, it's meant to be quite a low-cost solution that can use e.g. old DSTV satellite dishes for gathering signals from space.

There are three types of nodes:

1) MPICC gather node
2) MPICC hub node (which could incorporate the 'gather' functionality as well) and
3) MPICC processing centre. The processing centre doesn't need to be anywhere near.

Consider There are ADCs in the gather nodes. The gather node has an 'imager' component that performs sweeps to sample data the field of view (i.e. the part of the sky the antenna is looking at). You can consider the antennas are tuning in to signals in the frequency range 500MHz – 2GHz but only a band (a few KHz, up to 1 MHz) within this frequency range will be down converted and sampled.

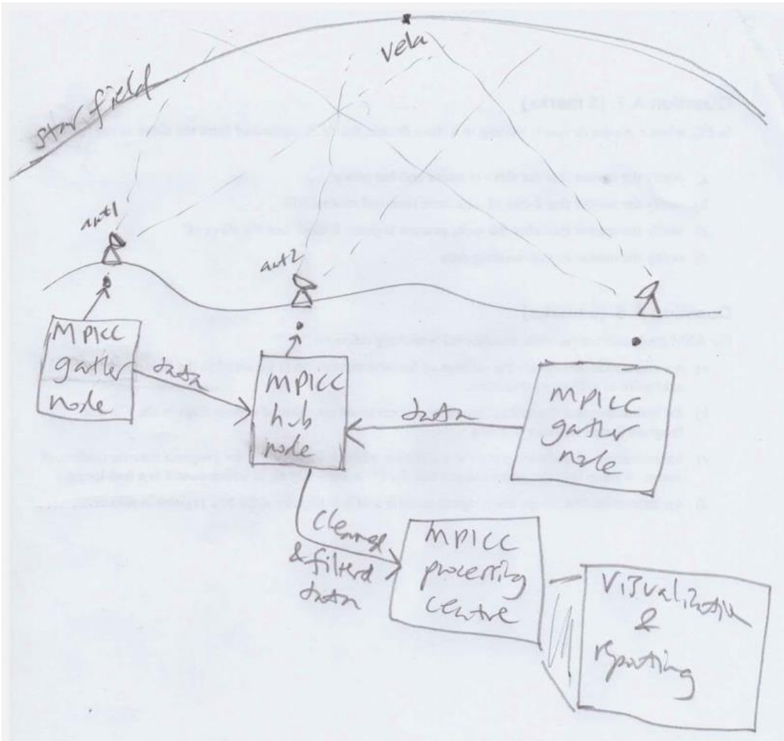


Figure 1: Sketch of the MPICC system, showing the different types of MPICC notes, their relation to antennas used in the mobile antenna array configuration.

The system is planned around the gather nodes sampling signals, to which pre-processing is done, timestamps and orientation information stored for each captured 'frame'. An 'interesting frames sets' is a collection of frames, that have the same orientation and receive frequencies, and which may contain a 'Wally'. We are considering a 'Wally' to be a potential pulsar in this assignment.

Q 1: **[GA aspect: 3]**

Let's do a quick bootup diagnostic test…. The topic of noise can have a significant influence on a ground-based radio astronomy telescope (it's generally know, and you surely know this too, that a radio telescope is not really aligned to the traditional definition of a telescope having optical lenses, but operation is abstractly speaking nevertheless very similar since there is much focusing and magnification of signals). But the radio frequencies of interest for radio telescopes do behave a bit differently to light. Considering this point, and that we're on a planet that generates quite a bit of heat among other things, and that a target of interest would be somewhere in the sky, does the direction of an antenna make much difference to the amount of noise that is picked up? Motivate your answer (whether you're right or wrong is not as important, in terms of marking, as the reasoning you provide).

ANSWER [3]:

Yes! The direction of the antenna does influence the amount of noise that is
picked up. It is generally 'sky noise' that has a significant influence on radio
astronomy. If there are sources of noise on the ground or in the sky (such as
satellites), especially if they send out strong signals as satellites and close-by
radio broadcasts do, but in this case the angle of the antenna still influences
the amount of noise picked up, e.g. if the position of the target intersects
these sources noise. Furthermore, it also depends on how much sky the
signals are travelling through. Pointing right up (i.e. antenna parallel to ground)
there is generally least sky noise, but pointing the antenna at lower angles
means more sky noise interference.

SCENARIO RELATED TO Q2.1 [GA], Q2.2 [non-GA], and Q3 [GA]

The gather node is closest to an antenna, where the sampling is happening. Consider that the gather node is to sample at 1Mbyte per second. The design team has been tasked to decide whether to use an Arduino, indeed the slightly fancier one with the 32-bit ARM Cortex-M0+; or the Raspberry Pi 4 with the, decidedly fancier, Broadcom 64-bit quad-core ARM Cortex-A72. But there's pros and cons in deciding which of these platforms for go for. Think of yourself on the design team for this project, that you have each platform, and that you have been tasked to decide strategies for benchmarking these platforms.

The processing to consider in regards to this question: Consider that you are capturing signals at 1Mbyte/s. The system is set up to centre frequency fc. That the dwell time on a coordinate is 100us, after which a radio image pixel is generated. Each pixel is an array of bytes showing intensity of frequency fc-16*S … fc … fc+15*S received. Consider S is the width of a frequency bin. So if S=20 Hz and fc=100MHz then we have a sampled sub-spectrum of 99,999,680Hz to 100,000,300Hz. The system is configured to do a sweep of 200 x 200 coordinates to generate one observation frame (or radio image) that is then sent to the hub node.

Q 2.1: **[GA aspect: 6]**

Read the scenario above and answer the following sub-question:

> Explain briefly what an observation frame is. How much smaller would you estimate the observation frame to be, compared to the raw sampled data from which the frame is generated by a set of sweeps. [GA related, do try to answer this!] [6]

ANSWER [6]:

An observation frame is a collection of pixels. These pixels may be
multi-dimensional (possibly a discrete Fourier transform or frequency bins of an
FFT) for positions surrounding the target position. So instead of just pointing
the antenna at an azimuth ϕ and elevation θ, the antenna scans positions
around (ϕ,θ) and saving these as a bitmap of pixels. In a simpler case, which is
often used, each pixel is just a magnitude indicating the strength of signals at
the frequency of interest. Only a small number of positions around the
central target position would be sampled, considering the time it takes. Likely
just around a hundred or two positions, say a 200x200 matrix of pixels. It if
each pixel is just an 8-bit grayscale recording of the magnitudes, each frame is
Just 40,000 bytes (just under 40KB). The raw samples that are gathered to
compute each pixel would likely be much more: sampling at a 1MS/s for
A few milliseconds (say 2ms) to sample one point would be 2/1000 * 1MB =
2097 samples (let's say it's 8-bit samples) then multiply by 200x200 which
gives 83880000 bytes = 79Mb (very close to 80MB). Or you can work it out as
40,000 samples * 2ms / sample = 80 seconds *1MB/s $\rightarrow$ 80MB.
Thus, a frame of just 40KB that took 80MB (!!) of raw samples to generate.
So I would estimate around 0.05% smaller. And this a fairly small size (if quite
realistic) observation frame size, bigger ones have an even smaller ratio.

Q 2.2: **[non-GA aspect: 10]**

Read the scenario above and answer the following sub-questions. Now, consider that the development team leader for the MPICC has assigned you to recommend which of these mentioned platforms, the Arduino or the Raspberry Pi, should be used for the gather node. Answer these sub-questions:

(a) Part of your task is to decide on what to benchmark for these platforms and how to do so. Briefly explain what you suggest benchmarking to aid this selection process (NB: excluding any suggestions about ACPI which is the topic of (b)). Provide some suggestions for doing this (i.e. no code needed in this answer). [5]
(b) What is 'Average Cycles Per Instruction (ACPI)' metric. Explain what information this provides. How relevant would this be as a metric to use in doing comparative benchmarking between the two platforms? [5]

(answer on next page … use back of next page if you need more space)

ANSWER for 2.2(a) [5]:

For this question the student is expected to make mention of the usual types
of benchmarking aspects, such as processor speed, interfacing (and PIO)
facilities that the processors and boards provide. Would set up experiments
to do testing, a major one being sampling from the ADC and saving the data
to RAM. Then checking speed of processing the sampled data to see that the
processor can get frames generated sufficiently quickly. But it is not just about
sampling, it is also processing the samples to generate frames. So doing
timing of how long each platform takes to do e.g. an FFT on 2Kb of samples
would also be samples necessary. And then there is also the matter of sending
on the samples further in the network, to the hubs and data centre.
While I didn't ask for you to reflect on which would probably work better, I'll
Nevertheless share my view: The Raspberry Pi would outperform the Arduino
on many aspects, possibly with the exception of PIOs and ease of interfacing.
The Arduino would probably struggle to capture the data sufficiently fast and
especially in terms of having sufficient memory space to store the captured data
and do FFTs on it. The RPi would probably have no problems, and be happy to
entertain the user with screensavers or games while also doing the main work.

ANSWER for 2.2 (b) [5]:

To understand ACPI, you consider the Cycles Per Instruction (CPI) for the
processor. CPI is defined as:
CPI = Cycles / Instruction Count
You would need to determine the CPI for each instruction used in the program
Expressed as CPI_i and the number of these instructions in the program, I_i.
Then using this data you can calculate the average CPI (ACPI) as:
$\text{CPU time} = \text{CycleTime} * \sum_{i=1}^{n} CPI_i * I_i$
How relevant would this be as a benchmark for comparing the RPi and Arduino
for the above scenario? Well, probably more difficult and less useful to do than
just running desired functions (e.g. FFT) on the platforms concerned.
The relevance would be more useful if the processors were different, if the
The Arduino selected had an AVR instead of an ARM (which some of the lower
cost version do have) then there would be merit. But otherwise it would be
comparing an single core ARM on the Arduino to a quadcore ARM on the RPi,
kind of comparing a few apples to more apples, so doing a ACPI comparative
analysis in this case would be entirely extraneous. (note to marker: if the
student is aware than indeed the two platforms might have the same
Instruction set and express futility of an ACPI analysis for such a case, then that
should get towards 100% 5/5 if not a bonus mark).

Q 3: **[GA Aspect: 10]**

Review the scenario above and respond to the following sub-questions.

(a) The gather node is described to be 'sweeping' around a point of interest. But why are we looking at coordinates slightly around a coordinate of interest? As opposed to just dwelling the whole time on the same coordinate? Provide a well-reasoned motivation for your response. [6]

(b) Considering the data volumes for generating a 200x200 frame versus a smaller frame of 100x100 in which the area covered is smaller (smaller area surveyed around the target). Although this might not be desirable, discuss what sort of speed-up this adjustment might achieve for the system. (a broad but technically sound discussion is fine, no need for calculating specific values, although welcome to do so if you like). [4]

ANSWER for 3(a) [6]:

The sweeping is done because the Earth is rotating and due to potential
lensing and refracting that may happen to the signals on the way from the

source to the antenna. The antenna itself might also be effected by the
environment, for instance wind, rain or vibrations that could impacting the
angle of the antenna. Dwelling for the whole time on one point also has
drawbacks as it may provide too little data regarding the target of interest to
do an accurate identification of the source.

(use back of page for more space!)

ANSWER for 3(b) [4]:

Reducing the size by a quarter would certainly reduce the amount of data
that needs. Basically, a quarter the size. *But* it is not only on the amount of
data that is produced by the gather node… it is also the amount of time
that is spent capturing and processing the data. Indeed this aspect is the
most significant aspect, for example the Arduino could probably handle a
100x100 better than a 200x200 (although an AVR would be hard pressed to
deal with even 100x100). So the smaller size will provide a significant speedup
of generating ¼ the frames, so instead of 80MB of raw data being captured it
Would go down to
10,000 samples * 2ms / sample = 20 seconds *1MB/s → 20MB.
A big saving in both memory and data needs and data to send over the net.

Q 4: **[GA Aspect: 9]**

As mentioned, there are three types of MPICC nodes, namely the gather, hub and processing centre. Consider that this system could be either very smaller, as in one gather that is also a hub, and one processing centre. How would the processing demands and data transfer demands increase as the size of the system gets bigger. How might this impact the platforms chosen, if the objective is towards keeping costs low (but not necessarily keeping the nodes equivalent between a small scale and a big scale configuration). Note the system is also aimed towards suiting a community of uses, some of which may get together on an ad-hoc basic to do astronomy observations. Would any of the node intent for a small scale still be usable in a big scale setup. In order words, would users choosing a cheapest instance be excluded from participating in a big team of observers? (argue & motivate your case for can or for can't.)

ANSWER [9]:

Marking advice: the main purpose of this question is that the student is
to provide a logical argument to motivate whether or not a 'cheapest instance'
would exclude participants in a team observation. Essentially, it would come
down to things about how big are the frames (big ones, needing faster
processing, more memory and generating more data) which may exclude
participants, for instance if large frames were used and the methods for

integrating frames did not support frames of multiple sizes (which would likely
be the case as such integration would be more like correlation of one image
with a blurred image, which has limited benefit compared to correlation in
which the matrices correlated are of the same resolution).
Nevertheless, it depends ultimately on the hubs and the data center, if they can
both operate on data of different resolution and cases where the number of
inputs (frames) for an observation change, e.g. one observation using 5 gathers
and another using just 3 with the 2 low-end gathers still busy postprocessing
or streaming data to the centre node.

(use back of page for more space)

SOME QUICK TRUE/FALSE QUESTIONS – SELECT ONLY ONE ANSWER OPTION…

Q 5: **[GA Aspect. 8 x 2 = 16]**

Put a tick for either to indicate TRUE (T) or FALSE (F) for each statement below:

Statement	**T**	**F**
a. Considering broadcast frequencies, a thing that impacts radio astronomy, even a signal at 1MHz is not classified as being high frequency.	X	
b. If two gather nodes are sending N observation frames to a hub, the hub will always be sending 2xN observation frames to the processing node.		X
c. While we refer to Wally as a pulsar of interest here, it's more an issue of spotting a star whose brightness cycles between fading and brightening over a certain period … as opposed to star wearing a red and white jersey.	X	
d. All pulsars are actually a group of at least two stars, i.e. a binary star system, where the pulsar effect is caused by the stars orbiting each other.		X
e. If you point an antenna directly down at Earther, then you'll basically eliminate all the noise picked up. (Although that obviously would not be so useful if you are trying to observe the stars).		X
f. Radio astronomy for finding pulsars needs to be done using more than one antenna, since a pulsar's light cone might be missed if using just one antenna.		X
g. The environmental humidity levels for a radio observation done at 1GHz and beyond is a factor of noise to be considered (i.e., below that level humidity has less impact on radio reception strength and potential refracting/multipathing).	X	
h. A whip or simple dipole antenna is the typical antenna used by serious radio astronomers because it is omnidirectional and thus has no need for manual focusing or orienting of the antenna.		X

Q 6: **[14]**

You've hopefully looked over the slides on effective bandwidth (Lecture 11). Read over the scenario that follows and answer the questions.

Scenario: You're working for a company that is providing internet access for ferries operating in an archipelago. Some ferries use satellite, or if seldom far from costal base-stations, they do not really need to connect with an independent WAN and can use the mobile phone network. However, the company concerned provides a WAN for shipping lanes. This utilizes many-channel microwave transceivers stations (or WTS) posted at optimal locations along the shipping lanes ferries use. When considering the worst-case situation, a ferry may be 50Km from the nearest WTS in foggy conditions. Microwaves scatter more in heavy fog, but still travels around 3x10^8 m/s. The ferry's WTS is equivalent to the ground-based WTS stations. A WTS transmit overhead takes 20us (these are quite directional, albeit wide-beam and may overlap but the receive side receive the highest power transmission on a given band). The receive overhead alas takes a fair bit longer, 200us to various checks and correlations on a sensed transmission, compensating best as possible for any interference of fading that may happen. The raw bandwidth for any one band is a rather impressive 400Mbit/s. For this question assume bytes will be sent as nine bits, the first bit will be a start bit (0), the subsequent eight bits are the data bits 1 to 8.

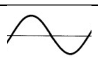
questions:

a) Calculate the effective bandwidth in bits per second for the 50Km distance from a ground WTS and a ferry WTS. [6]
b) What is the percentage efficiency for (a) above? Discuss your view on how efficient you think this connection is. [4]
c) If optimizing the receiver, by better utilize parallel computing and ultra-fast cache, the WTS receive overhead was cut down to 30us. What change would this make on the effective bandwidth. If it cost 1000 EUR to upgrade a WTS, and the company makes 1 EUR per 10Gb, discuss how much data would need to be transferred to recoup the investment. (Assume simplest case: 1 WTS and exclude cost of ferry WTS). [4]

ANSWER for 6(a): [6 marks]

Explanation here that the 3x10^8 is not so relevant to the problem

The talk about microwave scatter being more in heavy fog and leading to the microwave signals still travelling at pretty much around 3x10^8 m/s was partly just to see if students know the bigger picture of radio communications and signals and symbols or shaping. You should all already know this, but to put it simply, the symbols are shapes being formed, you can't really have one pulse at 1/(3x10^8) s immediately followed by another, those incredibly short bursts are one wavelength and probably get immediately lost in the clutter.

To form a shaped symbol like  takes a bit longer, you'd need to increase the amplitude over a certain period and decrease it. Same for / or \ shapes which would be less continuous and elegantly behaved than a sinusoid.

Anyway, there wasn't actually that much to do, if you read it you'd see:
see: 400Mbit/s. That's impressive. Sure, if you could send one bit as a
single 1/(3x10^8) s pulse, then maybe you'd get hyper speedy comms at
At 1/3 x 10^8 = 0.3T bps. That's not really going to happen, with sufficient
parallel channels, that could perhaps be reached.
The main answer is using the formula for effective bandwidth …
Total latency = Sending overhead + Transmission time + time of flight + Receiver overhead Effective bandwidth = Message size / total latency
It doesn't matter too much what the precise value is, it is more understanding
The use of the problem and applying the calculation effectively.

ANSWER for 6(b): [2 marks]

This would be Effective bandwidth / Raw Bandwidth
The receive overhead is going to dominate for short messages, although most
likely a ferry load of people are going to be demanding a fair amount of data
so the system would likely optimize streaming of data which would reduce
The weighting of the send overheads. But it would come down to a certain
amount of dependence on the protocols and how the channel is being used.
For streaming this would be efficient, it depends much on the protocol, but
this is not something expected of the student to discuss. The main problem
Is that with that slow receive overhead, which may lead to inefficient use of
the channel particularly with one send and transmit per vessel; for many small
messages particularly much of the time may end up being dominated by the
received overhead.
Marking notes: nevertheless, whether or not the student discusses these
specific aspects or not is not essential, the main objective is inspecting
adequate GA achievement of providing a logical and suitably motivated
response to this question.

ANSWER for 6(c): [4 marks]

Optimizing the WTS by that extent (30 instead of 200 us, speedup of 6.67) will
much reduce latencies in receiving messages, particularly in cases of short
messages and may lead to better multi-user utilization of the channel.
An argument could be formulated in that a saving of 170us time is achieved,
the system can send 400 000 000 bits per second so how many bits in 170us…
well, that is 170 / 1000000 → 0.00017s thus 68 000 bits, but it's 9bits per data
byte, so actually 7555.56 bytes or 7.38Kb. So, it looks like it will need a lot of
time to make up that cost.
1 EUR / 10Gb 1000EUR to upgrade one WTS would need
10000 Gb to be transferred
10000 x 10^9 = 10^13 bytes needing to be transferred… that is looking
rather unfeasible to attempt to recoup that investment just on saving a few
microseconds in this application and costing model.

Q 7: **[16]**

Answer the following sub-questions:

a) Edge computing is a field of computing that has shown much research and innovative innovations in the past decade, and the relevance of HPES towards such innovations was a point highlighted early in this course. But what is edge computing and why would HPES have any relevance to these technologies? [6]

b) Why is there potential merit in developing a parallel algorithm or solution, instead of remaining with sequential ones that are generally so much easier to understand? [6]

c) Surely a correctness proof is just another name for product validation. If you think this is true, just say True and move on (maybe I'm being nice 😉). But if you think otherwise, explain why a correctness proof is not necessarily entirely appropriate as a means for doing product validation. [4]

ANSWER for 7(a): [6 marks]

Edge computing (a precise definition) is a distributed computing paradigm that
brings computation (and possibly data storage) closer to where it is needed, as
a means to improve response times and save bandwidth. [from lecture 3].
HPES has much relevance to these technologies as it is able to do more
processing in a smaller formfactor at lower power utilization than is the
case for more traditional HPC systems. Furthermore, an emphasis around
HPES is close connection with sensing or signal acquisition and/or outputs that
links directly to a high-performance processing facilities, not necessarily

needing interleading networking infrastructure. Hence, HPES has been shown
a means to provide high-performance computing at the 'edge'.

ANSWER for 7(b): [6 marks]

This was intended to be a quick and easy question for you. Of course there are
many potential merits in developing a parallel algorithm or solution because
it could lead to faster operation, doing more processing at once, achieving
product development goals, enhancing the reliability and accuracy of
computation (e.g. running multiple algorithms to check answers are consistent)
but indeed that may still come at a cost of having more complicated code.
Although, there are cases where the parallel code could be easier than the
sequential code, such as cases where you can just run a simple algorithm on
multiple processors, using different inputs i.e. Single Program Multiple Data
(SPMD) style, instead of trying to change the code so that it will load in different
data and put it in different places.

ANSWER for 7(c): [4 marks]

No, indeed 'correctness proof' is NOT just another name for product validation;
it is rather about mathematical models or model checking that proves that
a process operation (explained as a mathematical model) will achieve
the correct response for any designed operational cases; this is a method
of mathematically proving the operation. It gets quite involved, and it tends
to be used for very critical applications where the system must operate
correctly in all cases. But, as can be reflected on in the story of the Mars
Climate Orbiter launch failed due to a human error of having mistakenly mixed
up feet and meters in operational parameters, which were probably fed in
to the correctness proof models as well, which would have led to an inaccurate
representation of reality in the models used for checking. Not sure if
ChatGPT would have helped to picked that up if it can check code.

Q 8: Multiple Choice [16]

Finally, last question (that sensible students might have done first!). Don't know who's happier about that, you or me 😉

Answer all multiple-choice questions below. Select only one answer option for each question or part of question. Note that Q8.2 and Q8.3 needs an answer and short motivation for that answer for both part i and part ii.

Q8.1 **4**

Which one of the following definitions sounds most accurate for defining what is meant by the concept of a Golden Measure as presented in Lecture 2 of this course.

(A) It's a 'yard stick' for measuring how fast your system should be.

(B) It's a special number, (1+sqrt(5))/2 to be precise, which is used in testing the accuracy of a machine's floating point arithmetic.

(C) It's a solution that may be slow but gives very accurate results.

(D) It's a (usually) unreachable perfect result, essentially a Golden Fleece of computing that a truly dedicated developer would sacrifice endless time and effort to provide.

Q8.2 **6**

Consider that you are on the team for building the 'Icarus Accelerator', which is an accelerator that uses DMA to access two blocks of memory, input block A and output block B. It also has a parameter, T (assume T<<N), which is the threshold for the number of non-black pixels (i.e. RGB value 0 0 0) wanted in a row to copy. Both A and B are NxN RGB full-colour images. At the start, A is loaded with an image, B is cleared. When the kernel runs it simply copies each entire row of pixels in A that have more than T non-blank pixels over to the corresponding row in B. Leaving B potentially with a couple of black lines that had few pixels in them. Hence, it's called Icarus Accelerator because only those lines with enough glue or interesting bits to process are kept aloft for further processing. But what sort of domain decomposition and level of granularity is going on here? NB: select an option for both parts i and ii below.

Part i	Domain Decomposition [2 marks]	Part ii	Granularity [2 marks]
(A)	Continuous	(A)	Very fine-grained 1:N*N (high dependence on all other data)
(B)	Blocked	(B)	Moderately fine-grained (each result dependens on much of other data)
(C)	Interlaced	(C)	Intermediate 1:1 (each result depends on its own data)
(D)	Cyclic	(D)	Moderately course-grained
	Motivation for this choice [1 marks]:		Motivation for this choice [1 marks]:

It's rows in an image, divided by each	*It isn't fine-grained because the processing*
line of the image or frame as per the	*for deciding pixels to copy needs less data*
definition for this.	*that even that pixel's own data.*

Q8.3 **6**

Vertical and horizonal load balancing is not the same thing. This question has a part i and part ii, the select the option in Part i relevant to vertical load balancing, and the option in Part ii relevant to horizonal load balancing. Provide a brief motivation at the bottom for why that form of load balancing achieves that characteristic.

Part i	Vertical load balancing [2 marks]	Part ii	Horizontal load balancing [2 marks]
(A)	This involves putting one computer on top of another, which saves space.	(A)	Simpler scalability and opportunity for wider parallel processing.
(B)	This allows simplified processor design albeit more of them.	(B)	Faster, but less redundancy as data is not split between machines.
(C)	This promotes greater redundancy by distribution of data between machines.	(C)	Requires a greater variety (i.e. width) of processor designs in use.
(D)	This involves much dependence on one machine.	(D)	Needs less power than vertical as only one processor runs at a time.
	Motivate [1 mark]:		*Motivate [1 mark]:*
Vertical balancing is using more closely		*Horizontal balancing is using multiple*	
coupled processors, all running on one		*Machines that may or may not all have the*	
machines, typically with shared memory.		*same processor; simpler and more scalable.*	

END OF TEST