**EASTERN MEDITERRANEAN UNIVERSITY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CMSE/CMPE-443**

**Real-Time System Design**

**Midterm Exam**

**2023-2024 Fall Semester**

**November 11, 2023, 10.30**

**Name-Surname : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student ID : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Instructor:

Alexander CHEFRANOV

Time: 90 minutes

YOU CAN BRING THREE A4-SIZED SHEETS OF **YOUR OWN *HAND-WRITTEN* NOTES** TO THE EXAM. PHOTOCOPIES ARE NOT ALLOWED AND WILL BE COLLECTED.

**Calculators, telephones, and other electronic devices are not allowed**

**Try each question**. **Totally 4 questions, 8 pages**

**Grading**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | 1 | 2 | 3 | 4 | Total |
| Point | 25 | 25 | 25 | 25 | 100 |

**Task 1 (25 points)**

1. (5 points) What the real-time system (RTS) is?

The RTS is a computer system correctness of which is defined both by correct logic and meeting deadlines

1. (5 points) What is the hard RTS?

Hard RTS is such that a single deadline violation leads to the system failure

1. (5 points) What is the plant? How does RTS control the plant?

Plant is an object controlled by RTS. RTS controls a plant by getting information about its state from sensors, elaborating a control output that is fed to actuators affecting the plant.

1. (5 points) What are the sources of RTS inputs?

The sources of RTS inputs are sensors, and external devices

1. (5 points) What conversions are applied to RTS inputs/outputs? Why?

RTS inputs/outputs need conversion from analogue to discrete form and back, because sensor produce continuous outputs, and computers work with discrete data,

**Task 2 (25 points)**

1. (4 points) What is the event in RTS?

The event in RTS is the change of the natural control flow

1. (4 points) What is the periodic event?

The periodic event is an event that occurs periodically

1. (4 points) What is the periodic task, task period, task execution time?

The periodic task is a task invoked periodically. Task period is the time between two consecutive invocations. Task execution time is time required to complete the task.

1. (4 points) What is the formula for CPU utilization by a set of periodic tasks? Explain each variable used in the formula.

The formula for CPU utilization is , where and are the -th task execution time and period respectively, is the number of the task in the set.

1. (9 points) Calculate processor utilization for the following task set:

|  |  |  |
| --- | --- | --- |
| **Task#** | **Execution time** | **Period** |
| **1** | **1** | **9** |
| **2** | **2** | **11** |
| **3** | **1** | **5** |
| **4** | **1** | **6** |
| **5** | **1** | **8** |

**Task 3 (25 points)**

Consider 0-address machine instruction set below:

|  |  |
| --- | --- |
| **0-address machine (reverse Polish)** | |
| where the processor has a stack and some supporting hardware, at least a top of stack (TOS) pointer. | |
| **Operation** | **e.g. or comment** |
| load\_literal <int> | effect: TOS:=TOS+1; stack[TOS]:=<int> load a *constant* onto the top of stack; this can be used in arithmetic or to get an address onto the stack for use by a load or a store instruction later (it is splitting hairs to argue whether the literal is an address or a constant which might happen to be used as an address elsewhere) |
| load | effect: stack[TOS]:=memory[stack[TOS]] take the top-of-stack as an address, replace the top-of-stack with the contents of that address. |
| sto | effect: memory[stack[TOS-1]]:=stack[TOS]; TOS:=TOS-2 store contents of top of stack at the address in stack[TOS-1] then pop the value and the address |
| <opcd> | where <opcd> is add | sub |... effect: stack[TOS-1] := stack[TOS-1] <op> stack[TOS]; TOS:=TOS-1 |

For the expression

z=w/y-z

1. Give the reverse Polish notation (12 points)

Traverse the expression tree by LRN (left-right-node) strategy:

zwy/z-=

1. Write the code to implement it (13 points)

Load literal @y

Load literal @w

Load

Load literal @y

Load

Div

Load literal @z

Load

Sub

Sto

**Task 4 (25 points)**

1. (4 points) What is PAL? What is PLA? What is the difference between PAL and PLA?

PAL is programmable array logic

PLA is programmable logic array

The difference is that PAL allows programming AND junctions, whereas PLA allows programming both AND and OR junctions

1. (4 points) What is ROM? How many bits each memory cell shall have if it is known that the maximal value to store is 30? How many address bits are necessary if the number of memory cells is 5?

ROM is read-only memory expected for reading only, not for writing

A memory cell shall have 5 bits to accommodate 30

The number of address bits shall be 3 to address 5 memory cells

1. (4 points) How can ROM be realized as PLA? What will be used as its inputs and outputs?

ROM can be realized as PLA if it realizes a set of Boolean functions that outputs necessary memory cell content when its input is the address of that cell represented as a sequence of bits.

1. (4 points) What is the number of PLA inputs and outputs to implement ROM keeping 5 numbers each less than 30?

The number of PLA inputs is 3 and the number of PLA outputs is 5

1. (4 points) Fill in the table below by the binary equivalents of the following five decimal numbers: 3, 1, 17, 21, 5

|  |  |
| --- | --- |
| Decimal Number | Binary number |
| 3 | 00011 |
| 1 | 00001 |
| 17 | 10001 |
| 21 | 100101 |
| 5 | 00101 |

1. (5 points) Draw a PLA schema programmed by fusing (connecting by dot) lines to implement a ROM with 5 memory cells containing the numbers in the above table in the given order.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | X2 | X1 | X0 | Y4 | Y3 | Y2 | Y1 | Y0 | Value |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 17 |
| 3 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 21 |
| 4 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 5 |

Y4=x2’x1x0’Vx2’x1x0

Y2=x2’x1x0Vx2x1’x0’

Y1=x2’x1’x0’

Y0= x2’x1’x0’V x2’x1’x0V x2’x1x0’V x2’x1x0V x2x1’x0’

Hint: Example of PLA with 2 inputs and 4 outputs



&0

&1

&2

&3

&4

X2

X2’

X1

X1’

X0

X0’

Or4

Or3

Or2

Or1

Or0

y4

y3

y2

y1

y0