1995 Midterm Exam Solution

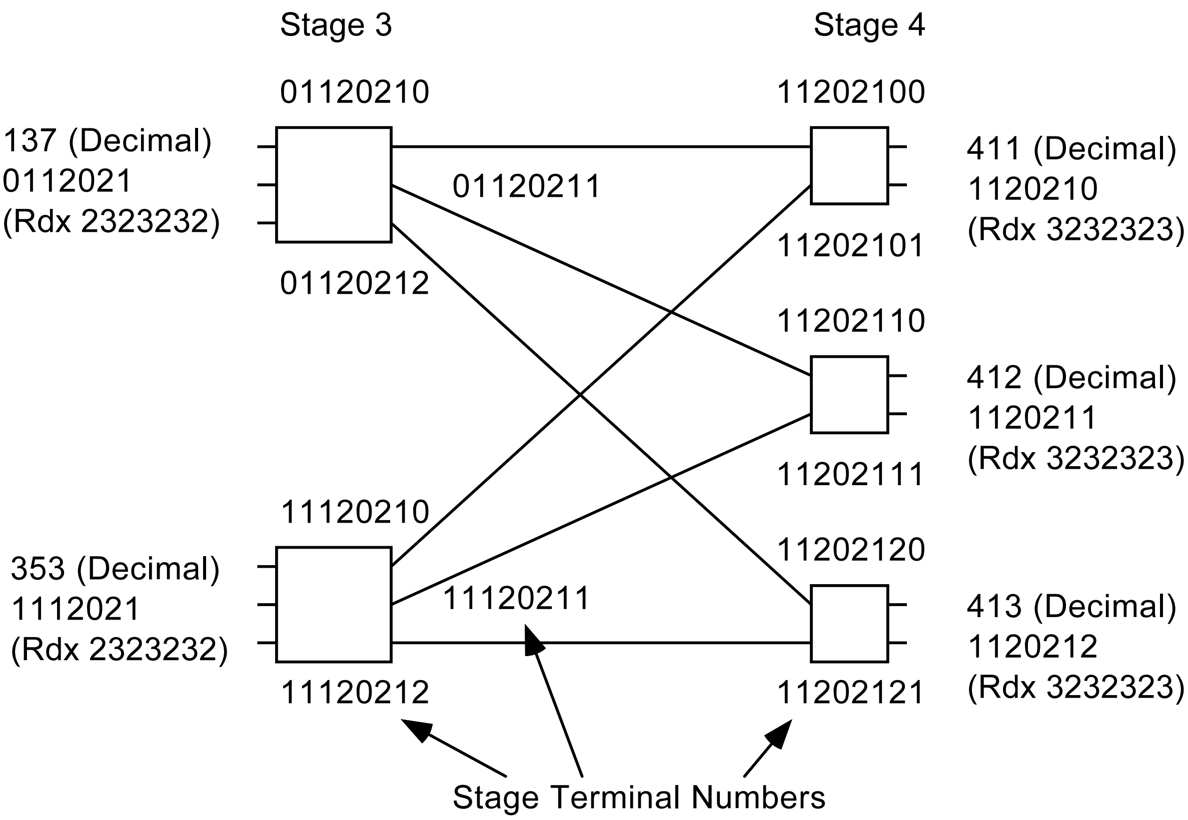
Overall class performance: Good!

Everyone did well on at least one problem.

Grade mean, 64.4; standard deviation 13.6.

Problem 1 Part a

Consider cell ⟨4, 413⟩ (cell 413 in stage 4) of a $(8,(2,3,2,3,2,3,2,3),-,\mathrm{S},\mathrm{T})$ omega network.



Red cells: $\boxed{\langle 3,\ 137\rangle,\ \langle 3,\ 353\rangle,\ \langle 4,\ 411\rangle,\ \langle 4,\ 412\rangle,\ \text{and}\ \langle 4,\ 413\rangle.}$

Problem 1 Part b

Prove that there are exactly five red cells.

Plan:

- Show labels of the five cells.
- Show that they are distinct.
- Show that no other cells in the two stages connect to these.

Cell Labels

Consider stages i and $i+1$ in which stage-i cells are 2×2.

Consider cells

$$\left\langle i,\, 0u_{(n-3)}u_{(n-4)}\dots u_{(0)}\right\rangle$$

$$\left\langle i,\, 1u_{(n-3)}u_{(n-4)}\dots u_{(0)}\right\rangle$$

$$\left\langle i,\, 2u_{(n-3)}u_{(n-4)}\dots u_{(0)}\right\rangle$$

$$\left\langle i+1,\, u_{(n-3)}u_{(n-4)}\dots u_{(0)}0\right\rangle$$

$$\left\langle i+1,\, u_{(n-3)}u_{(n-4)}\dots u_{(0)}1\right\rangle$$

Because the label of each cell differs in stage number, MSD or LSD, the cells are distinct.

Lemma: Each stage-i cell is adjacent to every stage-$i+1$ cell listed and no others.

Proof: By definition of omega network:

Cell $\left\langle i,\, xu_{(n-3)}u_{(n-4)}\dots u_{(0)}\right\rangle$ is adjacent to $\left\langle i+1,\, u_{(n-3)}u_{(n-4)}\dots u_{(0)}y\right\rangle$ for $x \in \{0,1,2\}$ and $y \in \{0,1\}$.

Each pair of adjacent-stage cells is in this form.

Every cell in these forms is in the list above.

Problem 2:

Part a:

$$V = \langle m^n \rangle$$

$$\begin{aligned} E = &\{\, (u, \sigma_{m,m^{n-1}}(u)) \mid u \in V \,\} \cup \\ &\{\, (u,v) \mid u \in V,\ u_{(0)} < m-1,\ v = u+1 \,\} \cup \\ &\{\, (u,v) \mid u \in V,\ u_{(0)} = m-1,\ v = u-(m-1) \,\} \end{aligned}$$

Part b:

Part c:

$$\delta = \begin{cases} 2, & \text{if } m = 2 \text{ and } n = 2; \\ 3, & \text{if } m = 2 \text{ and } n > 2; \\ 3, & \text{if } m > 2 \text{ and } n = 2; \\ 4, & \text{if } m > 2 \text{ and } n > 2. \end{cases}$$

Part d:

Consider nodes $(00\ldots0)$ and $(\left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m}{2} \right\rfloor \ldots \left\lfloor \frac{m}{2} \right\rfloor)$.

There are n digits.

$\left\lfloor \frac{m}{2} \right\rfloor$ links needed to change each digit in node label.

$n-1$ links needed to move between digits.

Diameter, $d = n \left\lfloor \frac{m}{2} \right\rfloor + n - 1$.

Problem 3

Part a:

Values of $p > 0.6$.

The queues will fill up no matter how large they are.

Messages must wait there turn so...

Increasing queue size increases latency.

Part b:

Each PE has about $\frac{P}{N}$ threads.

Each thread can have at most one message in the network.

Therefore, the maximum number of messages in the network is fixed.

Increasing queue size, after a point, will not result in fuller queues.

Increasing queue size will also reduce full-queue blocking.

Therefore, the performance will increase.

Part c:

Count crosspoints for each crossbar.

Number of crosspoints is product of number of inputs and outputs.

Total is $2mb$ crosspoints.

Part d:

Middle-buffered better:

- Message A cannot proceed; message A can fit in a middle-buffer queue, but is to large for an input-buffer queue.

- Message A cannot proceed; a second message, B, can use the same input. Message B would be blocked in an input-buffered cell.

Input-buffered better:

- There are b messages blocked, each going to a different output and each in a different queue. Other messages cannot proceed. They would be able to proceed in an input-buffered cell.

Part e:

Note that since program is frequently modified, new traces would be frequently needed.

Trace-Driven Better:

- Programs take a long time to run. It's therefore better to run them on a real system and collect a trace, even though they would be used only a few times.

- Programs require a realistic environment that execution-driven simulator cannot easily provide.

Execution-Driven Better

- Trace file would be too large.

- Network timing would effect program execution.

- Simulated system has special features so program must be run on simulated system.