Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl.-Ing. Christoph Baumann

Exercise Sheet 10
Computer Architecture II
(Due: Jan 24th, 2012)

Exercise 1: (Fast Half Decoder) **(4+4+4 Points)**

An n-bit half decoder is a circuit with inputs $x \in \mathbb{B}^n$ and outputs $y \in \mathbb{B}^{2^n}$ adhering to the following specification.

$$\langle x \rangle = i \iff y = 0^{2^n - i} 1^i$$

We already constructed half decoders with linear delay. Now we search for a faster implementation.

a) Let $n = 2^m$ be a power of two. Construct a half decoder with logarithmic delay!
Hint: Looking into fast decoders might be helpful to solve this exercise.

b) Prove the correctness of your implementation!

c) Show that it in deed has logarithmic delay!

Exercise 2: (Receiver Optimization) **(4+4 Points)**

The *strobe* circuit in the receiver contains a 3-bit incrementer and a 3-bit equality tester for comparing the value of the 3-bit counter with 4. For such a small number of inputs it may not be optimal to use the standard implementations for incrementers and equality testers. The exact cost and delay has to be determined.

a) Construct a circuit which compares any bit string $x \in \mathbb{B}^3$ with the binary representation of 4 and checks for equality. Prove its functionality! Cost and delay shall be $C_{3eq4} = 5$ and $D_{3eq4} = 3$. Compare this to the exact cost and delay of a 3-bit equality tester! Points will be deducted for sub-optimal solutions.

b) We already know that incrementers can be implemented in a carry-chain or conditional-sum style, where the first has linear delay and cost and the latter has logarithmic delay and linear cost. Which design should be chosen for the 3-bit incrementer if we are aiming for low delay, or low cost respectively? Compute the exact cost and delay for both implementations!

Our cost and delay model is listed in Table 1.

Gate	NOT	NAND/NOR	AND/OR	XOR/XNOR	1-MUX
Cost	1	2	2	4	3
Delay	1	1	2	2	2

Table 1: Cost and delay for common gates and circuits

Exercise 3: (Lemma 6.6 Base Case) **(10 Points)**

In this exercise you need to show the base case of Lemma 6.6 that was introduced in the lecture.

With the assumptions

- for all byte indices $i \in [0 : L-1]$, and
- for all bit indices $y \in [0 : 9]$ if $i < L-1$ and $y \in [0 : 10]$ if $i = L-1$

we claim:

a) receiver synchronizes in the correct cycle range:

$$sy(i+1) \in cy(8 * (3+10i)) + [3 : 4]$$

b) receiver votes the correct bit:

$$v^{str(3+10i+y)} = f(m)_{3+10i+y}$$

c) receiver strobes the correctly voted bit:

$$str(3+10i+y) = sy(i+1) + 8y + 4$$

d) receiver steps correctly through automaton until the next sync:

$$z^{str(3+10i+y)+1} = \begin{cases} BSS1 & : \quad y = 0 \\ b_0 & : \quad y = 1 \\ \vdots & \\ b_7 & : \quad y = 8 \\ BSS0 & : \quad y = 9 \wedge i < L-1 \\ FES & : \quad y = 9 \wedge i = L-1 \\ TES & : \quad y = 10 \wedge i = L-1 \end{cases}$$

The proof of this lemma is done over sync intervals i. Now prove the claim for $i = 0$!