University of the Saarland
Department 6.2 - Informatik
Prof. Dr. W.J. Paul

Computer Architecture I - WS 02/03

(due: 11.11.2002)

Excercise 1: (Compound Adder) **(5+3+2 points)**

In figure 1 you see the recursive Definition of a compound adder with inputs $a[n-1:0]$ and $b[n-1:0]$[1]. This circuit computes simultaneously $s^0 = \langle a \rangle + \langle b \rangle$ and $s^1 = \langle a \rangle + \langle b \rangle + 1$.

1. Compute cost and delay of a n bit compound adder in a closed formula[2]. Cost and delay of the basic gates (AND, OR, XOR, XNOR) are 1. Cost of a n bit multiplexer is $3n$. Its delay is 2.
2. Proof the correctness of your formulas by induction.
3. Compound adders can be used to construct conditional sum adders. Give a recursion construction (draw a figure) of a n bit conditional sum adder which uses compound adders.

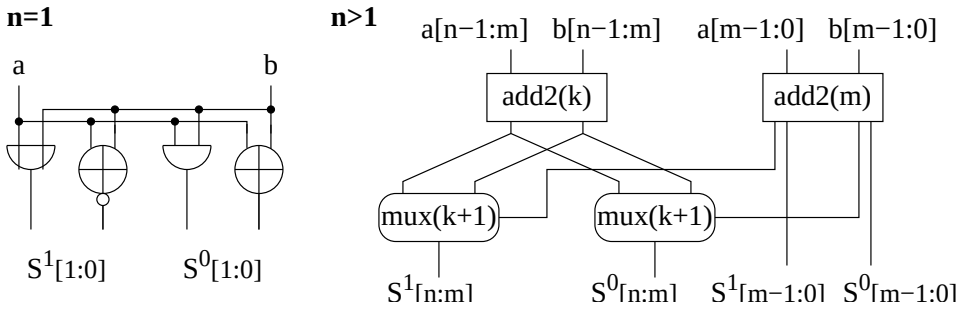

Figure 1: Compound Adder

Exercise 2: (Cyclic Left Shifter) **(3 + 3 points)**

For string $a[n-1:0]$ and natural numbers $i \in \{0, \ldots, n-1\}$ we consider the functions

$$\begin{aligned} cls(a,i) &= (a[n-i-1] \ldots a[0] a[n-1] \ldots a[n-i]) \\ crs(a,i) &= (a[i-1] \ldots a[0] a[n-1] \ldots a[i]) \end{aligned}$$

The function $cls(a,i)$ is called a *cyclic left shift*, the function $crs(a,i)$ is called a *cyclic right shift*. For a constant i we can construct a cyclic left shifter by the construction show in figure 2. The input i is called the shift distance of the shifter.

1. Give a construction of a n bit cyclic left shifter for arbitrary i with $0 \leq i < n$. This shifter should use $log(n)$ cyclic left shifters with fixed shift distance.

[1] Gates with a point at their lower end are negated. e.g. For $n = 1$ S^1 is computed by an OR and a XNOR gate.
[2] You can assume that n is a power of two.

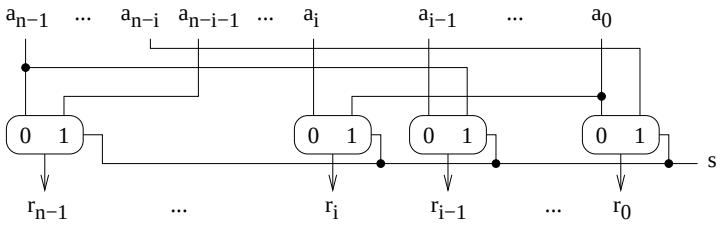

Figure 2: Cyclic Left Shifter for a constant i

2. How can you build a cyclic right shifter using a cyclic left shifter? Why is your solution correct?

Exercise 3: (Binary Representation) **(3 + 3 points)**

1. Show that there is only one binary representation for a given number a (if one cuts away leading zeros).

2. Let a and b be n bit binary numbers: $a, b \in \{0,1\}^n$.

   Let $<_{lex}$ be defined by:

   - $n = 1$: $a <_{lex} b \iff a[0] = 0 \wedge b[0] = 1$
   - $n > 1$: $a <_{lex} b \iff a[n-1] <_{lex} b[n-1] \vee (a[n-1] = b[n-1] \wedge a[n-2:0] <_{lex} b[n-2:0])$

   Proof the following claim:
   $\langle a \rangle < \langle b \rangle \iff a <_{lex} b$.

Exercise 4: (realistic cost model) **(8 points)**

For this exercise we will use the following cost and delay model:

- Components are placed on a regular grid (see figure 3).

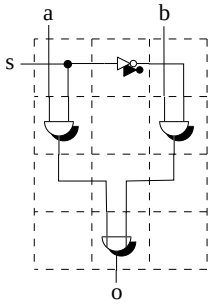

Figure 3: Grid Cost Model

- A simple gate has size 1x1: it uses one cell of the grid.
- Connections between gates also use cells. Connections can leave cells at the upper, lower, right or left side, but not at the corners.

- The cost of a circuit is the number of cells which are used by gates and connections of the circuit.
- The delay of a simple gate is one. The delay of a connection cell is also one. The delay of a circuit is the maximum delay between some input and some output of the cicuit.

For the example in figure 3 (a one bit multiplexer) we have cost 9 and a delay of 7.

Find a circuit for an n bit OR tree[3]. Give a recursive definition of your circuit. Design your circuit for speed (small delay). The faster your circuit is the more points you will get.

[3] An OR tree with input $a[n-1:0]$ outputs a 1 if any of the input bits is 1: $o = a[n-1] \vee \ldots \vee a[0]$.