Saarland University
Department 6.2 – Computer Science
Dr. M. A. Hillebrand
Prof. Dr. W. J. Paul

Computer Architecture I – WS 06/07

Exercise Sheet 4 **(due: 20.11.06)**

Exercise 1: (protocol design and control) **(4+4+4+4+4=20 points)**

Consider two circuits, a master and a slave, with the master posing requests to the slave and the slave responding to the requests. Master and slave are connected with the following interface signals:

- a request signal $req \in \mathbb{B}$ controlled by the master.
- a busy signal $busy \in \mathbb{B}$ controlled by the slave.
- a data input $in[n-1:0] \in \mathbb{B}^n$ controlled by the master.
- a data output $out[n-1:0] \in \mathbb{B}^m$ controlled by the slave.

The protocol itself is specified informally in the following way:

In order to start a request, the master raises the req signal. The slave will lower the $busy$ signal eventually to acknowledge the end of the request. During a request, the request signal req and the input bus in have to be kept stable. At the end of the request, the slave will return certain data on the output bus out.

1. Formalize these protocol conditions for the master and for the slave separately.
2. Arbiter with Priority:

   Design a clocked circuit A, which allows to connect two masters m_1 and m_2 to a single slave s. The interface of circuit A consists of a set of interface signals for all modules, i.e., for $x \in \{m_1, m_2, s\}$ we have signals $x.req$, $x.busy$, $x.in[n-1:0]$, and $x.out[m-1:0]$. In case that both master start a request simultaneously, the request of master m_1 should be preferred. Moreover, the circuit should not delay a request, i. e. 0-cycle or 1-cycle requests (depending on the slave) should still be possible.
3. Prove that your circuit guarantees that all protocol conditions are satisfied at all interface $x \in \{m_1, m_2, s\}$, given that the master fulfill the master protocol conditions (they keep certain inputs stable) and the slave fulfills the slave protocol conditions (it responds eventually). For your proof you will need the additional assumption, that the request signal of master m_1 is *not* always activated. Additionally, prove that request of master m_1 have priority over master m_2.
4. Fair Arbiter:

   In the arbiter with priority it is possible, that a request of master m_2 is never serviced, if master m_1 always poses a requests. Construct a fair arbiter, which guarantees that any request by master m_i for $i \in \{1, 2\}$ is serviced after at most one intervening request of master m_{3-i}. Again your construction should not delay requests.
5. Prove the correctness of your construction.

Exercise 2: (general purpose register) **(10 points)**

In class we have seen the informal specification of the general purpose register (GPR) for the DLX processor. In this exercise you are asked to construct a general purpose register. Hence, construct a circuit that fullfils the following properties:

- The GPR should be based on a 3-Port RAM, i. e. your construction should have inputs $w \in \mathbb{B}$, Din, Aad, Bad, and Cad as well as outputs DoA and DoB.
- The address width should be 5, i. e. the three addresses Aad, Bad, Cad are in $\mathbb{B}^5$.
- The data width is 32-bit, i. e. Din, DoA, and DoB are in $\mathbb{B}^{32}$
- At read address 0 the data word 0^{32} is returned at the output and the address 0 cannot be written.

Hence your construction should suffice the following semantics:

$$DoA^t = \begin{cases} 0^{32} & \text{if } (Aad^t = 0^5) \\ gpr^t(Aad^t) & \text{if } (Aad^t \neq Cad^t) \vee (w^t = 0) \\ \text{unspecified} & \text{otherwise} \end{cases}$$

$$DoB^t = \begin{cases} 0^{32} & \text{if } (Bad^t = 0^5) \\ gpr^t(Bad^t) & \text{if } (Bad^t \neq Cad^t) \vee (w^t = 0) \\ \text{unspecified} & \text{otherwise} \end{cases}$$

$$gpr^{t+1} = gpr^t \text{ if } (w^t = 0)$$

$$gpr^{t+1}(a) = \begin{cases} Din^t & \text{if } (a = Cad^t) \wedge (Cad^t \neq 0^5) \\ gpr^t(a) & \text{otherwise} \end{cases} \text{ if } w^t = 1$$