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

Computer Architecture II - SS 04

(due: 21.09.2004)

This exercise will be discussed on Tuesday, 21.09.

Excercise 1: (Fair Scheduler for the CDB) **(10 + 15 + 5 points)**
Let n be the number of functional units. The Common Data Bus (CDB) is requested by the producer of function unit i by raising the $FU_i.CDBreq$ signal. The CDB control environment activates exactly one $FU_j.CDBack$ signal during the next cycle. Let $R(t)$ and $A(t)$ be defined in the following way:

- $R(t)$ is the set of all producers which request the CDB in cycle t:
  $R(t) = \{i \in \{0, \ldots, n-1\} | FU_i(t).CDBreq = 1\}$
- $A(t)$ is the set of active acknowledge signals in cycle t:
  $A(t) = \{i \in \{0, \ldots, n-1\} | FU_i(t).CDBack = 1\}$

In each cycle t multiple producers might be requesting the CDB. The CDB control has to choose exactly one because only one unit can use the CDB in cycle t. The correctness proof of the Tomasulo scheduling algorithm with reorder buffer requires a guaranty that any unit requesting the CDB will get an acknowledgement within a finite limit of time. This is done by allocating the CDB *round robin.*

In this excercise you have to construct a circuit $CDBcntrl$ which computes from the request signals $CDBreq(t)$ and the *old* acknowledge signals $CDBack(t)$ the *new* acknowledge signals $CDBack(t+1)$.

Let i be the number of the functional unit which was acknowledged in cycle t. Thus in cycle $t+1$ $CDBcntrl$ has to acknowledge that requesting functional unit j with $j = min\{k | k > i \wedge CDBreq_k(t)\}$. If no such j exists (i.e. no functional unit with number greater than i requests the CDB) $CDBcntrl$ has to acknowledge the requesting functional unit with the lowest index.

1. Construct a circuit $CDBcntrl_1$ which gets as input the old acknowledge signals $CDBack(t)$ and computes as output $Y[n-1:0]$ where $Y[j]$ is active iff j is greater than the index i of the functional units acknowledged in cycle t. (Hint: it may help you to use a parallel prefix circuit.)
2. Use the circuit $CDBcntrl_1$ and a $2n$ bit Find First One circuit in order to compute the number of the functional unit that has to be acknowledged in cycle $t+1$.
3. What does your circuit compute in case no functional unit is requesting the CDB in cycle t. Is this behaviour correct?

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

Computer Architecture II - SS 04

(due: 21.09.2004)

Definition of Find First One circuit:
A Find First One circuit ffo computes the following function:
$ffo : \{0,1\}^n \rightarrow \{0,1\}^{n+1}, (a_{n-1}, \ldots, a_0) \rightarrow (b_{n-1}, \ldots, b_0, zero)$, such that b_i is active iff $i = min\{j \in \{0, \ldots, n-1\} | a_j = 1\}$. $zero$ should be active if and only if $a_i = 0$ for all $i \in \{0, \ldots, n-1\}$.

Excercise 2: (Overall Scheduling Example) (15 points)

Take the following example scheduling of two instructions and fill the tables on page 2 with the correct values.

We consider the following instructions:
I_1: `R3 := M[R10]`
I_2: `R1 := R3 + R2`

For this example, `M[R10]` contains the value 11 and R2 contains 9. In cycle t=0 the first instruction is already in the execution phase. It is executed by the memory unit and stored in reorder buffer entry 0. Furthermore, in cycle t=0 the second instruction is fetched.

In cycle t=1, this instruction is decoded and issued into a ALU reservation station. The ALU is assumed to have only one reservation station. The reorder buffer entry 1 is also filled with this instruction.

In cycle t=2, the load instruction is on cycle ahead of completion. Thus, the memory reservation station requests the CDB for the next cycle.

In cycle t=3, this request is acknowledged by the CDB control. The result of the load operation (11) is put on the CDB. This makes the second operand of the ALU reservation station valid. Since both operands are now valid, the instruction is dispatched into the ALU in the same cycle. Furthermore, the ALU requests the CDB for the next cycle. In the same cycle, the result of the load instruction on the CDB is written into the reorder buffer entry 0, which becomes valid.

In cycle t=4, the result of the load instruction is written from the reorder buffer entry 0 into the register file. R3 becomes valid by this. In the same cycle, the CDB control acknowledges the CDB request by the ALU. The result of the addition is put on the CDB and reorder buffer entry 1 becomes valid.

In cycle t=5, this result is finally written into the register file.