

Computation Structures — Tutorial 4

September 29, 2015

μ -code for ULg03 – Virtual Memory

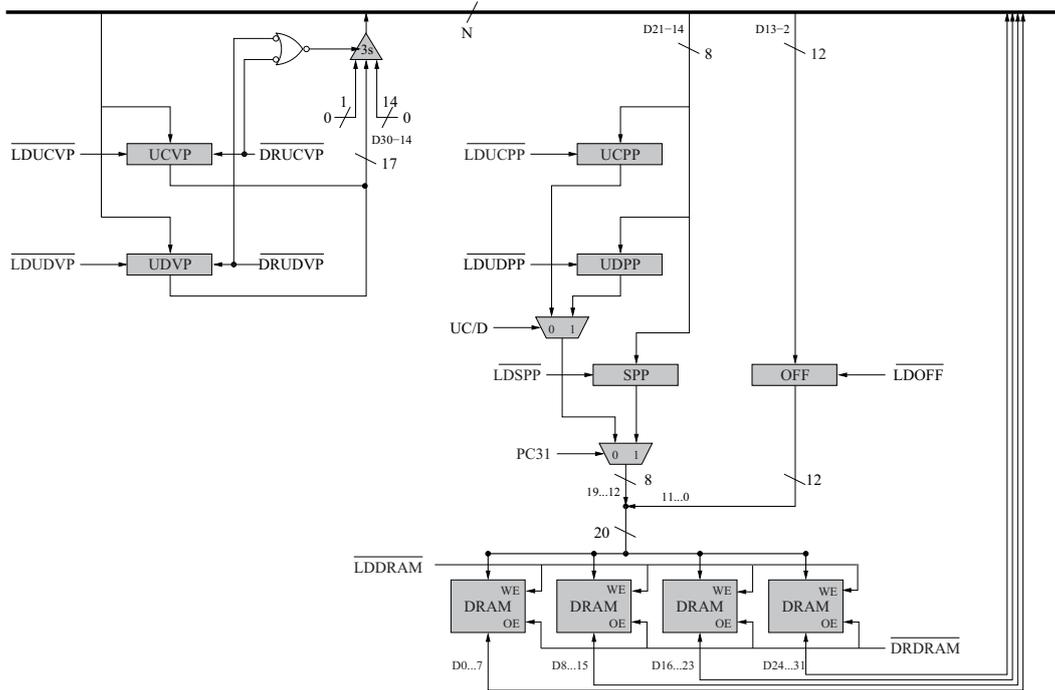
Reminder

- ULg03 introduces *virtual memory*. In practice, it translates into a more complex paradigm to access data and code stored in DRAM.
- With virtual memory, user programs think they have access to a large contiguous space of dynamic memory, while physically it is fragmented and partially on disk.
- Purpose of virtual memory: keep memory representation simple for user programs without impacting the performances. Indeed, reserving a fixed contiguous memory space for each process is excessively constraining for the system and the processes themselves.
- Summary of changes made to ULg02 to implement VM:
 1. New DRAM module (DMAR has been replaced by a new circuit).
 2. New control signals output by the control ROM:
 - LD and DR signals for UCVP, UCPP, UDVP, UDPP;
 - UC/D: 0 for code and 1 for data.
 3. Constants ROM updated with offset mask and the addresses to two new handlers (*cache miss code* and *cache miss data*).
 4. Micro-code can be longer (up to 32 phases).

Exercises

1. Provide the ULg03 user *and* supervisor micro-code for the BNE(Ra, label, Rc) instruction.
2. Combine LD(Ra, Lit, Rd) and JMP(Rd, Rc) in a single instruction. This new instruction JMPI(Ra, Lit, Rc) directs the program to an address found in memory at the address Ra + Lit. Register Rc shall receive the address of the instruction immediately following the JMPI we're executing. Provide the ULg03 user *and* supervisor micro-code for the JMPI(Ra, Lit, Rc) instruction.
3. We wish to upgrade ULg03 DRAM module so that the machine memory is 8MB instead of 4MB. There are two ways of doing that; indicate them on the provided schemas. Explain briefly.

Le module RAM dynamique de ULg03



Le module RAM dynamique de ULg03

