Midterm Th in class OB, DN, NS
CP1 - done!
CP2 - cache (no interrupts)

Exam - HW, Labs/C81
K-maps
MIPS datapath
MIPS pipeline
ready/valid interface
UART
Verilog for simple FSM and FF+gates
RAM, ALU, reg datapath
FF internals
CLB internals

K-maps
from: Z(m or TM or f(x,y,z))
to K-map
to minimal SOP, POS
to implementation w/ nmos, pmos, etc.

MIPS datapath, single cycle
add some new funct. e.g. register inst.
MIPS pipeline
timing of inst., hazards, forwarding
clock period, tPD, tPG, tsetup
Ready/Valid
- draw traces
- errors

UART
- timing, FSM

Verilog for simple FSM
- e.g. "output a 1 when you see XYZ"
- state transition diagram
- verilog FF & gates
- define states, state resets
- next state logic
- output logic

RAM + ALU + registers
- e.g. "find the largest/XYZ add store in register"
- draw datapath
- draw state transition diagram

FF internals
- know how they work down to FETS
- add funct.
- trace output given input XYZ

CLB internals
- I'll give you my detailed info that you need.

Registered PAL
Cache

why do we have registers?
- efficient instruction encoding
- faster than RAM
  - lower energy/ops

Registers are smaller, faster, lower energy.

way to store variables.
Compile (or assembly code writer) deals directly
between registers & variables

Harvard vs. Princeton

Is your project Harvard or Princeton?