**HLSG: High Level Study Guide, subset presented by Ilia Lebedev**

**Introductory Material:**

Concept of Hierarchy in designs and example hierarchies.

- Please, please, please, please use this on tests. Try to grade your own diagrams.
- When your design has "a thing that does X" (everything with a definable interface and function), it should be a block in a block diagram.

Concept of cost/performance/power tradeoffs and simple examples.

Power-Performance-Cost. We optimize for one or two of these, at the expense of others. Technology defines limits.

"*pareto optimal*" = best tradeoff (highest possible performance given power, cost).

"*pareto optimal curve*" = graph best possible performance against cost. Same for power. Depends on underlying technology.

**CPU microarchitecture:**

Implementation of a single-cycle processor from ISA description.

Implement this processor down to the gate level:
http://highered.mcgraw-hill.com/sites/dl/free/0072467509/104653/PattPatelAppA.pdf

Exchange your solution with someone else in the class and carefully ***grade*** their work.

Processor pipelining: impact on performance, hazard types and resolution.

Data hazard

- RAW Read after write. Bypass data to be written without actually reading it.
- WAR Write after read Make sure to commit (affect changes to arch. state) in order,
- WAW Write after write (same) but in your CPU, WAR and WAW are not a problem. Why?

Control hazard Branch/jump resolution. Add delay slots to the ISA (if allowed), or add lots of hardware.

Structural hazard

- duplicate conflicted resource (dedicated adder in X stage for branch evaluation).
- make the resource sharable (add a port to the register file to read out 2 values).
- stall or "inject a bubble", taking multiple cycles in one stage.

Detailed operation of 3-stage MIPS pipeline (read the course slides for lectures 10, 11 twice).

Serial communication and UART basics (read lec 12).

Memory mapped CPU I/O and polling implementation.

Using some addresses to control I/O (no memory at those addresses, the load is used to control the device). Lab5.

Polling means the CPU executes instructions to query the state of I/O (loads from the memory mapped io device)

**FPGAs**

| | |
|---|---|
| Idealized FPGA architecture model. | (read lecture 3, pages 5 through 13 twice) |
| Details of basic FPGA PIP (programmable interconnection point). | (same) |
| LUT implementation details. Hierarchical LUT designs. | (same) |
| Partitioning logic circuits into LUTs and CLBs. | (same) |

- The key idea is ***ANY*** combinational circuit with4 inputs and one output can fit into a 4LUT, no matter how big.
- Hardware can be duplicated (mapped to several LUTs (inverter in lec3, p13)

High-level details of Virtex 5 FPGAs (as presented in class).

- you may find Xilinx Xcell issue 59, p8-11 is a good summary. If not, redo the datasheet readings.

**Timing**

Relation of clock speed to performance.

- Performance formula (clock rate, program, and µarchitectural throughput all matter)

Determination of maximum clock frequency from circuit.

- Minimum period is determined by the **single slowest path from register to register**. It is the max of:
  - input register hold times (this dominates in structures like the shift register)
  - $\sum\{T_{Clock\rightarrow Q}, T_{through\ combinational\ \ logic}, T_{setup}, T_{uncertainty\ due\ to\ clock\ skew\ and jitter}, T_{safety\ margin}\}$

Origin of logic delay, Origin of flip-flop delay, Wire delay and mitigation.

- Read lectures 17, 18 twice.

Effects of clock skew

- Clock wires have delay. Minimize this delay by making better clock wires (H-trees), or add a safe margin to the critical paths.