ECE 720 – ESL & Physical Design

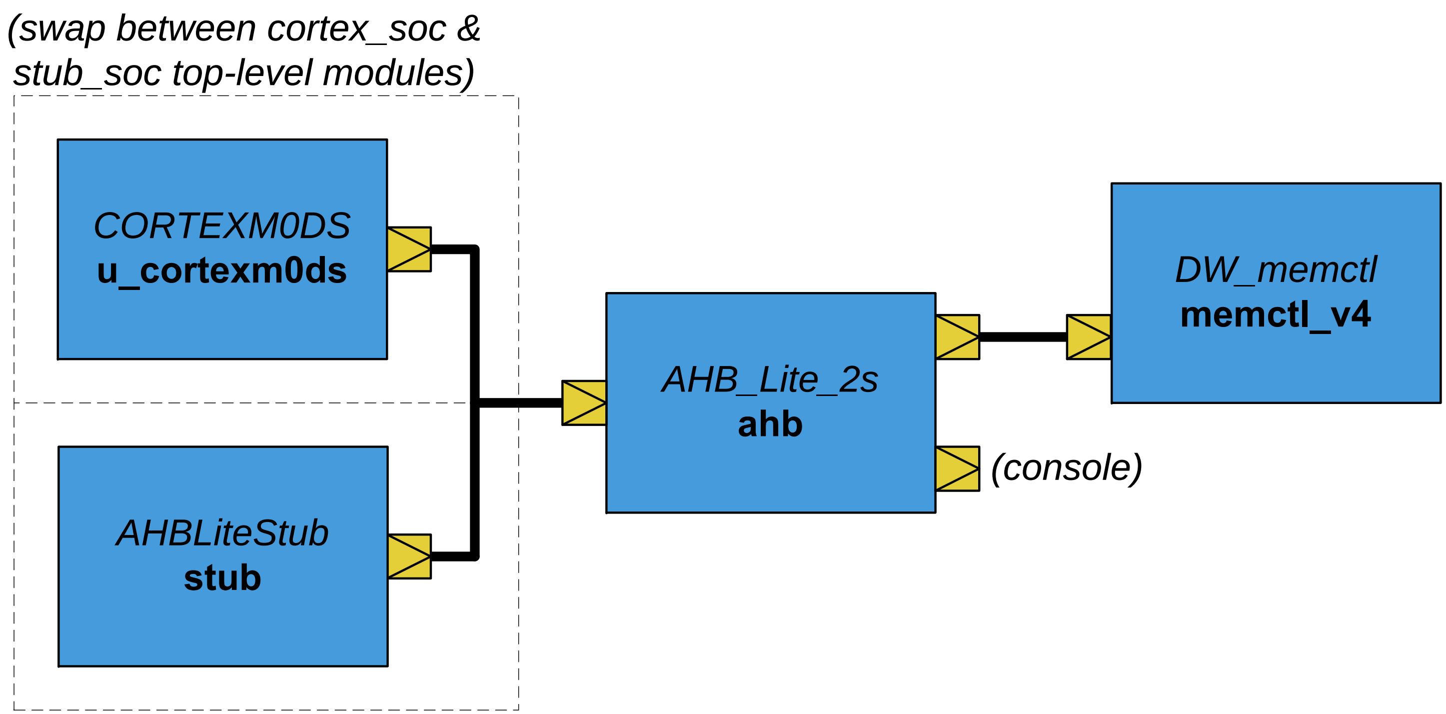
Project 1 Requirements

W. Rhett Davis
NC State University

Announcements

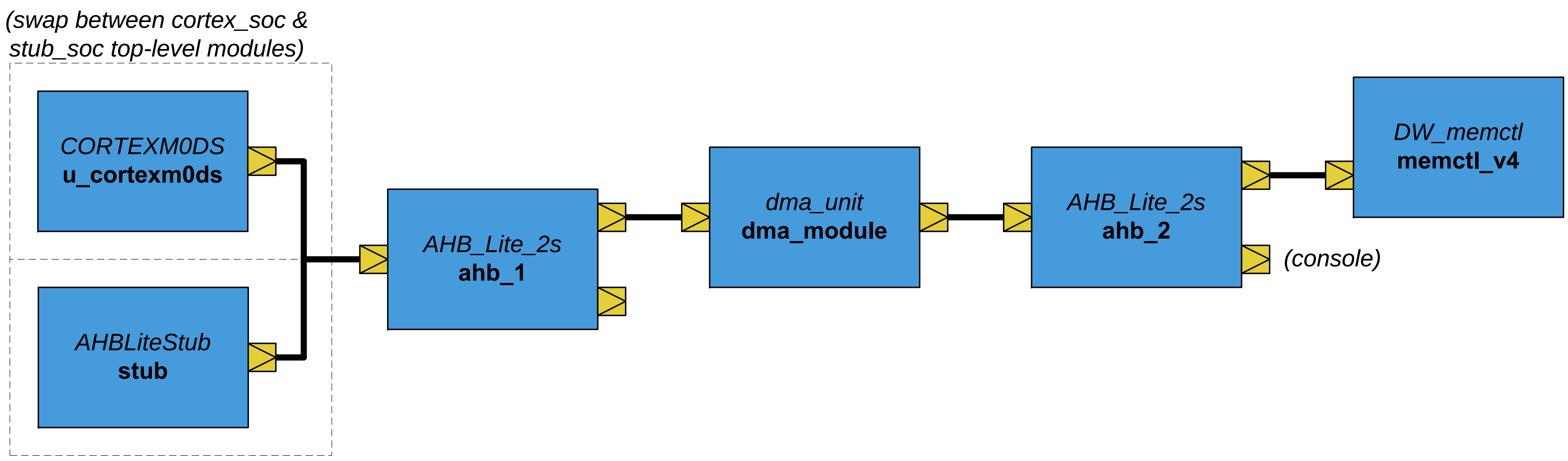
- Homework 2 due Thursday
- Project 1 Due in 2 weeks

Project System 1



- ***cortex_soc*** system includes CPU, bus, and DW_memctl
- ***stub_soc*** system includes stub in place of CPU
- **USE_STUB** macro in tb.v can be used to select stub system

Project System 2



- ***integrated_cpu_dma*** system includes CPU
- ***integrated_stub_dma*** system includes stub in place of CPU
- **USE_STUB** macro in tb.v can be used to select stub system
- All transactions are passed through DMA to ahb_2 except writes to DMA registers

DMA Registers

- SR - 0x40000010 – Source Register – Address of the first location to copy from
- DR - 0x40000060 – Destination Register – Address of the first location to copy to
- LEN - 0x40000090 – Length Register – Number of 4-byte words to copy
  Writing begins transfer

- Thanks to Reshma Siddalingadevaru for designing this DMA controller

Example Program (dma.c)

```
int main(void) {
  int i ;
  int *dma_sr = (int *)0x40000010;
  int *dma_dr = (int *)0x40000060;
  int *dma_len = (int *)0x40000090;

  for(i = 0; i < 10; i++) {}

  *dma_sr = 0x00000040;
  *dma_dr = 0x00004000;
  *dma_len = 0x0000000f;

  for(i = 0; i < 300; i++) {}

  . . .
```

- Transfers 16 words from 0x0040 to 0x4000
- Transfers 8 words from 0x0010 to 0x1000
- For loops used to insert delay (works around a bug that breaks simulation)

Slight change to Makefiles

- For Homework 2, SRCDIR variable was changed in sim/Makefile to swap between RTL and gate-level simulations

- Now, different Makefile targets are used
  - make sim, make gui for RTL
  - make gatesim, make gategui for gate-level

Project Goals

- Analyze the speed (*i.e.* cycles per transfer) and power (averaged over all transfers) of the SoC when using the DMA Controller for memory transfers
- Compare the speed and power of the SoC with and without the DMA Controller
- Predict how much the speed of the DMA controller could be increased with some re-design effort
- Find and diagnose bugs in the DMA controller
- Write a python script to automate your analysis

Requirements (1/3)

- Due Thu. Sep. 22
- Gate-Level Simulation Constraints
  - Do not modify the Verilog code in the src/gate directories
  - Create Three test-cases (*i.e.* C programs or transaction files) for the DMA controller system and determine the cycles-per-transfer and average power for each case
  - Create Three more test-cases for the system without the DMA controller and determine the cycles-per-transfer and average power for each case

Requirements (2/3)

- Gate-Level Simulation Constraints
  - Modify v/sim/Makefile so that “make sim” generates waves.vcd and the transaction-dump for one case that includes the DMA controller.
  - The “make sim” recipe must also execute a python script that computes and prints the number of read and write transfers for this test-case.
  - Modify v/synth/Makefile so that “make ptpx” predicts gate-level power for the same case above

Requirements (3/3)

- Documentation Requirement
  - Modify the provided compare_power.xlsx file to include the values for each case compared in your report
- Report Requirements
  - Values claimed in report must match values simulated with submitted code and compare_power.xlsx file
  - Report should contain an introduction, body, and conclusion. You may use whatever section titles you like for the body.
  - Report should contain at least one plot of power vs. transfers. All test-cases should be included in a plot.
  - Report should include one or more suggestions about how much faster the DMA controller could be and why you think so.

A Word About the Report

- Your paper is my only “window” into your project beyond the one case recreated with your code.
- Your final conclusions may be excellent, but if the report is poorly written, superficial, and not comprehensive, I will not be able to fully appreciate how good it is.
- Originality is important. If your test-cases and analysis look identical to everyone else’s, then the outcome of your project is less significant.

Project Grading

- (15%) Completeness & Organization – All requirements must be met
- (15%) Writing Quality – Emphasis on Clarity. Good grammar is necessary, but not sufficient
- (15%) Analysis – Graphs and explanation of results (15%)
- (15%) Execution – Ability to re-create a data-point
- (40%) Key Outcomes – awarded as follows:
  - (10%) No successful analysis of behavior, but significant effort
  - (20%) Limited range of behavior analyzed
  - (30%) Satisfactory range of behavior analyzed
  - (40%) Good and bad qualities of the DMA controller fully analyzed
- Recall that Project 1 is 20% of the total grade