

University of Pennsylvania
Department of Electrical and Systems Engineering
Electronic Design Automation

ESE535, Spring 2011

Assignment #3

Monday, February 14

Due: Monday, February 21, beginning of class.

Resources You are free to use any books, articles, notes, or papers as references. Provide citations in your writeup as appropriate.

Collaboration You may give tutorial assistance on using OS, compiler, and debugging tools. All code development should be done independently. You may **not** share code or show each other code solutions. All writeups must be the work of the individual.

Writeup Turn-in assignments on blackboard. See details on course web page. No hand-writing or hand-drawn figures. See details below on what you need to turn in and the format.

Assignment 3 Task Develop and implement an algorithm to schedule a circuit netlist, C , onto a time-multiplexed, programmable substrate:

- meeting a specified bound on the number of netlist nodes assigned to a PE
- achieving a target makespan (time-bounded scheduling) accounting for inter-cluster net delay
- while minimizing the energy required to evaluate the netlist C :

$$E = E_{inter_cluster} \times \sum_{PEs\ p} (p.inputs + p.outputs) + E_{intra_cluster} \times \sum_{Blocks\ b} (b.inputs) \quad (1)$$

We provide `compute_energy` in the updated `mesh.c` to perform a final evaluation of the energy cost. $E_{inter_cluster}$ is typically much larger than $E_{intra_cluster}$. Both are provided as argument inputs to `sched_main`.

- Unlike assignment 2, the number of used PEs is not a direct goal, but the energy function will typically encourage the use of fewer PEs.

For this week, we model the fact that it is faster and lower energy to communication within a PE. We charge a fixed unit of additional time to move a signal between PEs and a fixed unit of additional energy to move data out of and in to a PE. As a result, this problem adds a clustering component to the scheduling problem. Aside from the change in delay and energy costs, the architecture is unchanged from Assignment 2. The intra-cluster energy component will be set by the graph, so there is nothing you can do to change it; it is the inter-cluster energy that you have the option to control.

Code Base Changes:

Pickup the code in `assign3.tar` from `~ese535/spring2011/assign3.tar` on `eniac`. Changes have been made to `mesh.c`, `check_precedence.c`, and `schedule_main.c`, and `asap.c`, as well as the associated header files.

Notably:

- `check_precedence` has been updated to account for the additional delay required to move data between PEs.
- `compute_energy` has been added to compute the energy of a mapping.
- `legalize_schedule` has been added to take a placed design and reassign the timesteps to make the schedule legal (obey sequentialization within the PE and precedence constraints). This was added so that both the previous `asap` scheduler and your assignment 2 scheduler could be used with the new delay model.

Please keep your assignment 2 scheduler for comparison and add a new scheduler for this assignment. The test `Makefile` is setup to assume your assignment 2 scheduler is option 2 and your new scheduler is option 3. The new `sched_main` is setup to run `legalize_schedule` after you assignment 2 scheduler in option 2. You should update the option 3 case to call your new scheduler.

Comparing your assignment 2 scheduler result to your assignment 3 scheduler allows you to quantify the benefits of directly addressing the clustering goal versus ignoring it.

Looking forward, you will be addressing the communication distance for nets (placement) in assignment 4.

Turnin: You will need to upload two files. We have created separate assignments on blackboard so that you only need to submit a single file to each assignment.

- **assign3-writeup:** a single PDF with
 1. [**warmup: suggest you do this immediately and definitely before developing your algorithm.**] What is the energy saving for each of the following:
 - (a) Assigning two blocks that share the same input to a PE?
 - (b) Assigning a block to the same PE with one of its predecessors?
 - (c) Assigning all the blocks associated with a net to the same PE?
 2. Description of your algorithm and the code that supports it.
 - (a) A high-level, English text description of your strategy. Use references if appropriate.
 - (b) Define any cost functions used by your algorithm.
 - (c) Provide pseudocode for your algorithm.
 3. Completed version of this comparison table with your energy results based on the energy costs in `test/Makefile`:

Design	Energy @ 2.0×ASAP delay					Energy @ 3.0×ASAP Delay				
	ASAP	assign2	assign3	$\Delta_A\%$	$\Delta_2\%$	ASAP	assign2	assign3	$\Delta_A\%$	$\Delta_2\%$
example	193					193				
e64	7570					7570				
s1423	5198					5198				
alu4-em4	33949					33949				
frisc-em4	59756					59756				
s298-em4	43438					43438				

$\Delta_A\%$ is the percent improvement of your assignment 3 algorithm over the ASAP baseline:

$$\Delta\% \text{ improve} = \frac{ASAP_Energy - Assign3_Energy}{ASAP_Energy}$$

$\Delta_2\%$ is the percent improvement of your assignment 3 algorithm over your assignment 2 algorithm:

$$\Delta\% \text{ improve} = \frac{Assign2_Energy - Assign3_Energy}{Assign2_Energy}$$

- **assign3-code:** a single tar file with your code (no binary files, but in an archive like the provided support so it can be unpacked and built; this means the make file should be updated to build the application with any additional source files you may have added)
 - run `make clean` in both the code and test directories
 - use `make assign3.tar` to create the tar file
 - test that you can unpack your `assign3.tar` and build and run tests from there before you upload to blackboard