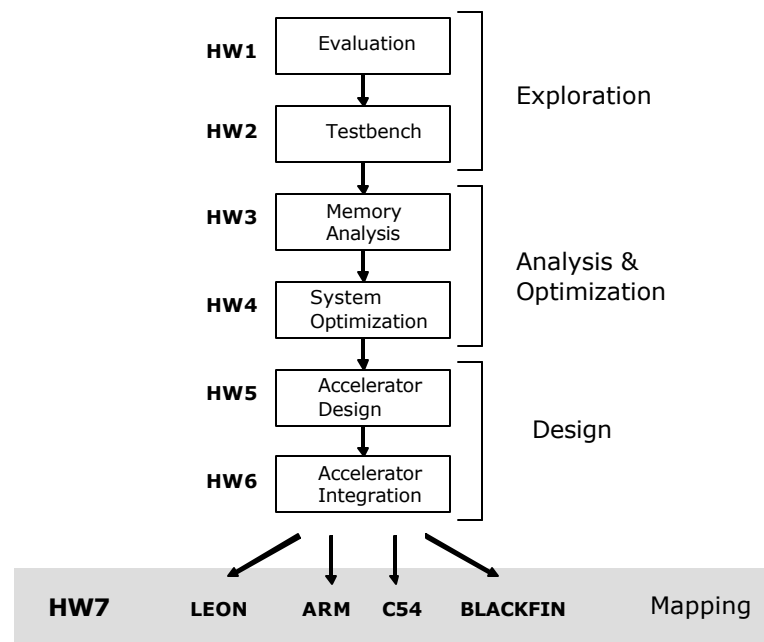


EE201A - Project Presentation

Mapping and Wrapping Up

Over the past quarter, we've taken several steps to bring a HTTP/TCP/IP protocol stack to implementation. Broadly speaking we had three phases in our project: (a) an exploration phase where we took a reference implementation and tested it in various ways, (b) an analysis phase where we considered behavior and optimization of memory access patterns, and (c) a design phase where we isolated and optimized one bottleneck in the protocol stack. While actual industry projects might be bigger, they follow essentially the same order of tasks.

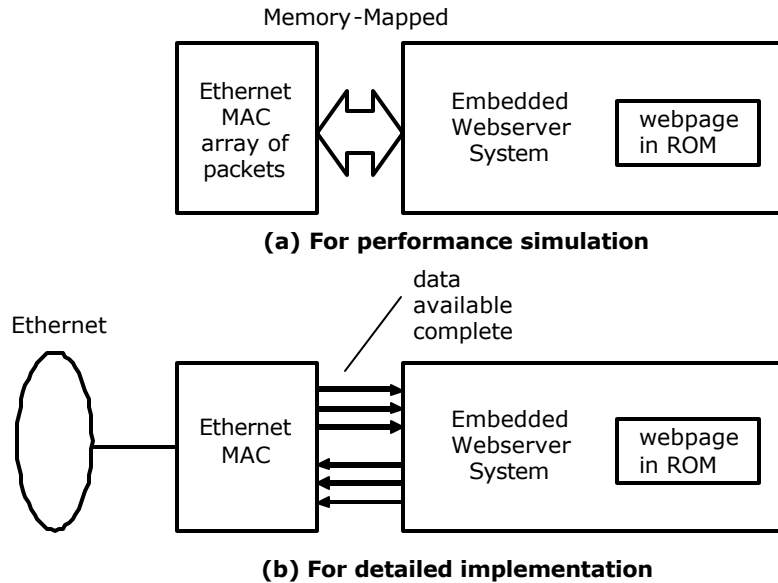


We are now ready for the final phase in the project: mapping to the actual target platform (LEON/ARM/C54/BLACKFIN). The goal of this phase is to create the implementation with the highest possible performance, using the techniques shown in this course. This includes transformations/optimizations at the system/ algorithm level, reworking the C code, and using assembly code or hardware acceleration for performance-critical parts. You can use any technique as long as you can show it is an evolution of your base target platform (LEON/ARM/C54/BLACKFIN). The target functionality includes IP/TCP and HTTP, as shown in the UIP code that we have used for Homeworks 2 through 4.

In contrast to the homeworks, there is no fixed 'recipe' by which you can obtain a solution. Making the best implementation of the Embedded Web Server is completely in your hands - you will make design decisions and prepare to defend them at the final presentation.

System Assumptions

You can assume that packets are delivered through a mechanism as used in HW6 (data, available, and complete). For coarse-grain performance simulations (using instruction-set



simulations, for example), you can make use of a memory-mapped interface (i.e. you store the input packets into an array, and you read them out of an array). For more precision, you need to make use of the interface protocol we discussed in HW6.

C54 or BLACKFIN

We already handed out C54 kits. We also have a number of BLACKFIN development kits available. Please come to lab 53-109 to pick up such a kit (if your target platform is BLACKFIN). You can install the kit onto a PC running Windows. We expect that you return all kits after the final project presentation and before the quarter ends!

If you are targeting DSP, then your primary method of performance improvement will be to exploit features of the DSP architectures (special computational units, Harvard architecture, special addressing modes and so on). Your measure of cost is code/data size of your program, your measure of performance is cycle count.

LEON or ARM

Your primary evaluation platform is TSIM (Leon) or ARM ISS in combination with GEZEL. In case of LEON, you can also consider implementation using the VHDL source code found at <http://www.gaisler.com>. You can also consider other ways of customizing the ARM or LEON platform (using the coprocessor interfaces, for example). These last two options offer a bigger, riskier challenge (with a potentially better result). Be realistic in your goals however.

In the case of LEON and ARM, you want to focus on how you can get the best performance out of a 32-bit RISC architecture, and what you can do to enhance performance further once you meet a bottleneck in software.

You can customize the system architecture in any way you see fit: adding custom memories, special processing units, and so on. The design goal is also here: go for the highest performance.

What to do

The end result of this project will be a presentation of your results during Finals Week. You can also summarize your findings in a report.

- *> By **Monday, June 9, Noon** at Marilyn's desk, you need to turn in your report. MAXIMUM length is 5 pages. Be formal and factual, indicate main results achieved (performance, cost factors like code size etc.) as well as the optimization methods used. Also, include your final code on your homepage.
- *> On **Wednesday, June 11**, the final presentations will be held. Specific time and location will be announced in class.
- *> Good luck, use the board to ask questions!

Also, in doing this final phase, be lazy and smart. Don't rush to your compiler or assembler, but think first. Take a step back and browse through the past results you have achieved in this project. Were there any particularly successful methods? Were there any obvious opportunities for optimization that you have not yet exploited? What can you do with your design to make it the single most performant and original implementation? Note also that all tools that we have used in the project are still available: ATOMIUM, SpecC, LEON/ARM GEZEL Cosimulators.