

The Story

The engineers at Golden Bear Circuits have been commissioned by Company X looking to do a low-cost video game project. You are on the team that is designing their chip, and you're working on an FPGA-based prototype. Company X is mostly a software company. They know how to write code to make great video games, but they need better performance than they've had in the past with software-based pixel manipulation on their MIPS-based hardware. They aren't finished designing their game, but they have given Golden Bear a list of requirements for the project (below).

You have an internal project review coming up, where you need to present the status of your work. People in the project review include your manager and his boss, and several of your colleagues on the project. Your colleagues are engineers who will demand technical detail to make sure that they understand what you're doing. For example, the people who wrote the Chrontel driver and the UART driver might be in the room. Your manager and his boss are not very technical at all, but need to understand that you are addressing the customer's requirements. You need to give a presentation that gives an overview of the project that your manager's boss will understand, but you also need to include enough detail that your colleagues can understand the engineering decisions that you made.

You want to both show off the good things that you've done, and honestly describe where you had problems and where you need help. Numeric metrics are better than qualitative assessments, e.g. "X pixels/second", or "Y pixels/cycle" are better than "really fast pixel drawing".

Requirements

Note that this is the exact list that was posted to the newsgroup, with the addition of point values in brackets in front of each requirement.

1) both moving and static objects

[10] 1.1) hitting spacebar pauses/un-pauses motion of objects

[10] 1.2) other keyboard input should control motion of at least one object

[20] 1.3) display time - there should be some mechanism for displaying time. analog clock, digital clock, you decide

[10] 1.4) borders around playing field - preferably dynamic. This keeps you honest about finishing drawing the frames on time!

[10] 1.5) ability to add objects from keyboard

[20] 2) must add at least one new command to the graphics processor. Examples include: line with varying color and/or brightness; triangle, circle, other wireframe object; filled triangle or other object; character or other pattern maps.

[20] 3) Your graphics engine must have a memory mapped control register, and use DMA to access graphics commands. Your keyboard input must use an interrupt (no

polling)

Presentation

During RRR week, you and your partner will do a 10 minute powerpoint presentation in the lab, in conjunction with a live demo of the game.

Things that should be in your presentation:

- * description of how you implemented everything. Start at the highest level block diagram, and work your way down. You won't have time to show it all, but the TAs and I are going to pick and choose what we ask for detail on. So have a few high level slides, and then have a bunch of backups that might not get shown. You should not include much/any verilog in your powerpoint.
- * description of the handoffs and tradeoffs between hardware and software. How did you implement erasing/redrawing the screen, motion, etc. What got done in "normal" C, what got done in ISRs, and what got done in hardware?
- * project performance evaluation. How many fixed and mobile pixels/frame could you handle? How many lines/frame? How many resources did you use, and for what?
- * self evaluation: what did you do well, what didn't go well?
- * future improvements: if you were to add more to the project, what would it be?

Make sure that between your powerpoint and game demo you make it very clear how you addressed each of the customer requirements.

Grading

C code is NOT a part of the grade. You should write only enough C code to demonstrate the hardware capabilities that you have created. There's no penalty for writing something fancy, but you won't get any extra credit for it.

Extra credit

- * display CPU and GP duty cycle
- * implement more than one additional graphics command
- * your clever hardware ideas