EECS 373
Introduction to Embedded System Design

Robert Dick
University of Michigan

Lecture 4: Debugging complex systems, APB

25 January 2024
Outline

- Memory-Mapped I/O Review
- Debugging Complex Systems
- Advanced Peripheral Bus
Example

```c
#include <stdio.h>
#include <inttypes.h>

#define REG_FOO 0x40000140

int main(void) {
    volatile uint32_t *reg = (uint32_t *)(REG_FOO);
    *reg += 3;

    print_uint(*reg);
    return 0;
}
```
"*reg += 3" is turned into a ld, add, str sequence

- Load instruction.
  - A bus read operation commences.
  - The CPU drives the address "reg" onto the address bus.
  - The CPU indicated a read operation is in process (e.g., R/W#).
  - Some "handshaking" occurs.
  - The target drives the contents of "reg" onto the data lines.
  - The contents of "reg" are loaded into a CPU register (e.g., r0).
- Add instruction.
  - An immediate add (e.g., add r0, #3) adds three to this value.
- Store instruction.
  - A bus write operation commences.
  - The CPU drives the address "reg" onto the address bus.
  - The CPU indicated a write operation is in process (e.g., R/W#).
  - The CPU drives the contents of "r0" onto the data lines.
  - Some "handshaking" occurs.
  - The target stores the data value into address "reg".
Outline

• Memory-Mapped I/O Review
• **Debugging Complex Systems**
• Advanced Peripheral Bus
This is quick and seems simple but it is actually deep and important.

A computer system is a graph.
Each component, e.g., a line of code or transistor, is a vertex (v).
Each effect that influences other components is an edge (e).
Complexity is a function of $|v| + |e|$.
Graph sizes

- For undirected fully connected graphs.
  - $|e| = |v|(|v| - 1) / 2$
- But it's much worse than that because your ability to analyze systems decreases dramatically with system size.
- So system complexity (debug time) is a superlinear function of $|v|$, like $|v|^k$.
- $k$ is generally $\geq 2$, and probably quite a bit bigger.
Best-case complexity
Worst-case complexity
Managing complexity
Incremental tested growth
Incremental tested growth
Incremental tested growth
Incremental tested growth
Incremental tested growth
Design and debugging: how to make your life easy and make your embedded systems work

- Control |v|
  - Get a very simple version of the system tested and functioning and add to it in small pieces, testing after each addition.
- Control |e|
  - Never build something big and then start testing.

- Build and test isolated, side-effect free components with narrow and easy-to-understand interfaces.
Start from the root

• Check the foundation before the roof.
• Check the power distribution network integrity before checking the software.
Switch between information gathering and reasoning

- Search process.
- Large search space.
- Probing specific locations is expensive.
- Initial conditions.
  - Don't have the data necessary to understand the problem.
  - Haven't done the analysis necessary to convert those data to information.
- Don't neglect either weakness. Iterate.
  - Conduct naïve experiments to gather information.
  - Stop testing and reason about problem, using conclusions to devise additional tests.
- Most engineers are better at analysis or testing.
  - Don’t stay under the streetlight.
Outline

- Memory-Mapped I/O Review
- Debugging Complex Systems
- Advanced Peripheral Bus
Details of the bus “handshaking” depend on the particular memory/peripherals involved

- SoC memory/peripherals
  - AMBA AHB/APB

- NAND Flash
  - Open NAND Flash Interface (ONFI)

- DDR SDRAM
  - JEDEC JESD79, JESD79-2F, etc.
Modern embedded systems have multiple busses
Advanced Microcontroller Bus Architecture (AMBA)
- Advanced High-performance Bus (AHB)
- Advanced Peripheral Bus (APB)

AHB
- High performance
- Pipelined operation
- Burst transfers
- Multiple bus masters
- Split transactions

APB
- Low power
- Latched address/control
- Simple interface
- Suitable of many peripherals
Bus terminology

Transactions have “initiators” and “targets”

• Only “bus masters” can be initiators.
  • In many cases there is only one bus master (single master vs. multi-master).

• Slave devices can only be targets. They can't start transactions, but they carry them out when a master initiates one.

• Some wires might be shared among all devices while others might be point-to-point connections (generally connecting the master to each target).
Driving shared wires

• Some shared wires might need to be driven by multiple devices.
• In that case, we need a way to allow one device to control the wires while the others “stay out of the way”
• Most common solutions are
  • tri-state drivers and
  • open-collector connections.
Another option: avoid shared wires

- Expensive when connecting chips on a PCB as you are paying for pins and wiring area.
- Doable but costs area and time on-chip.
Wire count

- Consider a single-master bus with 5 other devices connected and a 32-bit data bus.
- Shared bus → 32 pins
- Separate buses
  - Each slave would need ____ pins for data
  - The master would need ____ pins for data
- Pins and wiring area cost money.
APB is designed for ease of use

- Low-cost.
- Low-power.
- Low-complexity.
- Low-bandwidth.
- Non-pipelined.
- Ideal for peripherals.
Done.