Processors

Chapter 4

The Variety Of Processors And Computational Engines

Topics

- Introduction
- What Is A Processor?
- Range Of Processors
- Hierarchical Structure And Computational Engines
- Multiple Computation Engines
- Processor Categories
- Coprocessors
- Microcontrollers
- Microsequencers

Topics

- Embedded systems processors
- General-purpose processors
- Processor Technologies
- Stored Programs
- The Fetch-Execute Cycle
- Clock Rate And Instruction Rate
- Control
- Starting And Stopping
- Wrapup

Introduction

- Previous chapters
  - digital logic
  - data type representations
- Key elements in a computer
  - *processor*
  - memory
  - input and output

Introduction

- Topics covered in this chapter: *introduction to processors*
  - general concepts
  - variety of processors
  - relationship between clock and processing rate
- Topics covered in this section: *processors*
  - instruction sets
  - addressing modes
  - functions of CPU

What Is A Processor?

- Is processor = Central Processing Unit?
- Processor =
  - digital device that can perform a computation involving multiple steps.
  - computes boolean arithmetic, etc.
  - need not be extensive or powerful.

Range Of Processors

- Processor categories based in order of flexibility (least to most)
  - fixed logic
    - least flexible, single built-in operation.
  - selectable logic
    - several functions
    - exact function is determined at invocation.

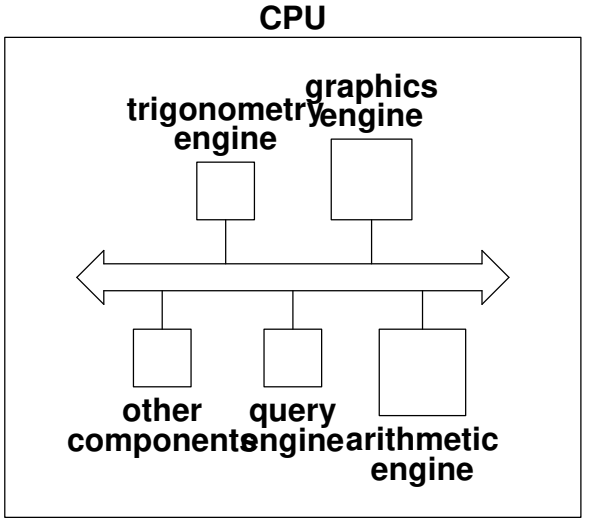
Range Of Processors

- parameterized logic
  - computes a predetermined function, but accepts input parameters.
- programmable logic
  - most flexibile
  - processor given a program to run, typically placed in the memory.

Hierarchical Structure And Computation Engines

- CPU's are highly complex
- Architects use hierarchical approach is used in design and testing
- Subpieces are called computational engines.
  - example of engines: trigonometry, query, graphics, pattern engines, etc.
  - each engine has a specific functionality.

Multiple Computation Engines



- An example of a CPU with multiple engines

Processor Categories

- Processors are used in variety of roles. Example
  - coprocessors
  - microcontrollers
  - microsequencers
  - embedded systems processors
  - general-purpose processors

Coprocessors

- Special purpose high speed processor that operates in conjunction/under the control of another processor
- Example, floating point accelerator
- Fixed or selectable logic processors

Microcontrollers

- Programmable device dedicated to controlling a physical system
- Example, automobile engine, elevators
- Performs trivial functions

```
do forever {
        wait for the sensor to be tripped;
        turn on power to the door motor;
        wait for a signal that indicates the
            door is open;
        wait for the sensor to reset;
        delay ten seconds;
        turn off power to the door motor;
}
```

- Example of steps a microcontroller performs

Microsequencers

- Microcontroller for internal control
- Controls coprocessors & engines within larger processor
- Example, microsequencer to control data movement

Processors

- Embedded Systems Processors
  - Runs sophisticated electronics
  - Example, DVD
  - More powerful processors than microcontrollers, more recent.
- General-Purpose Processors
  - Example, CPU of a PC

Processor Technologies

- Chronologically
  - digital logic circuits
  - LSI
  - IC

Stored Programs

- Processors performs series of steps in computation.
- Programs are stored either
  - internally in a few cases
  - in externally accessible location, e.g. main memory where it is easy to change.
  - ROM
- *Architect considers a processor programmable if processors are separate from program it runs*

Fetch-Execute Cycle

- Questions
  - how is the program represented ?
  - how does processor identify next step of program
  - what are the operations that can be performed and how does the
    processor perform them?

- Fetch-execute paradigm
  - programmable processor accesses 1 step of a stored program,
    executes it; repeat above with next step

Fetch-Execute Cycle

- *At some level, every programmable processot implements a fetch-execute cycle*

Repeat forever {

Fetch: access the next step of the program from the location in which the program has been stored.

Execute: Perform the step of the program.

}

- The algorithm used in the fetch-execute cycle.

Clock Rate And Instruction Rate

- How fast does the fetch-execute cycle operate? Depends on
  - processor (clock speed)
  - program storage technology
  - time to execute each instruction
- Fetch-execute cycle does not proceed at fixed rate.
  - example, multiplication takes longer than addition.

Control

- Questions with fetch-execute cycle,
  - how does a processor start running the cycle?
  - what happens after the processor executes the last step in a program?

Stopping and Starting

- Stopping
  - processor does not stop till powered down.
  - so what´s the next step to execute?
    * dedicated system: repeat the program
    * general purpose: run operating system programs when no user application is running

- Starting
  - depends on hardware.
  - possibilities: reset, bootstrap,soft power switch.

Wrapup

- Processors are digital devices that perform multiple step computations
- Categories based on roles: coprocessors, microsequencers, microcontrollers, embedded systems, general purpose.
- Programmable processors
  - hardware separate from stored program.
  - run fetch-execute cycle
  - processor must have some instruction to execute, never stops.