Outline (Part 1)

• Definitions and Examples
• Hardware Engineering Challenges
• Embedded Processor Examples
  - TI MSP430F2001
  - TI OMAP DM3730
  - Atmel ATtiny4
  - Xilinx FPGAs/MicroBlaze/Zynq-7000
  - Microsemi IGLOO FPGA

• Closing Comments
• Preview of Part 2 (April 22nd)
Embedded System

Historical Definition

• A dedicated computer performing a specific function as a part of a larger system

• Saw commercial use beginning in the 1970s as an alternative to hard-wired control and logic circuitry (engine controls, guidance systems, industrial process controllers)

• Essential Goal: Turn hardware problems into software problems.
Embedded System
Modern Definition

- High-reliability systems operating in a resource-constrained environment (typically cost, space & power)

- Excludes general-purpose computers, and non-computerized devices (now rare!)

- Smartphones are computers, but technical challenges are the same (ES may not be a very meaningful term any more, since it applies to so much of the ECE/CS universe!)
Are ES Software or Hardware? “Yes”

- No externally-visible difference between hardware and software functions

- Well-defined, fundamental, and extremely performance-sensitive functions are generally implemented in hardware.

- Complex, non-performance-sensitive, and/or subject-to-change functions are generally implemented in software.
Embedded Systems Examples

**Consumer**
- Cellular Handsets
- Game Consoles
- Media Players
- TV Set-Top Boxes
- Kitchen Appliances

**Vehicular**
- Engine Control
- Driveline Control
- Anti-Lock Braking
- GPS Receivers
- Tire monitoring “Infotainment”

**Infrastructure**
- Utility meters
- Traffic signal control
- Structure monitoring
- Surveillance
- HVAC control
- Parking meters
- Battery Management
- Generator controls
- Pipeline Safety

**Medical**
- Pacemakers
- Measurement
- Drug Delivery

**Industry**
- PLCs
- Motor Control
- Process Control

**Defense**
- Vision Enhancers
- Communications
- UAV Systems

**Computing**
- (computers within computers!)
Ford EEC
"Old School"

- **EEC-I (1978, about same time as Apple II)**
  - Proprietary 12-bit processor @ 3 MHz
  - "PM-11" after the DEC PDP-11
  - 4K bytes of ROM, 256 bytes of RAM
  - Ignition timing, EGR valve & smog pump

- **EEC-IV (1983 through 1990s)**
  - Intel 8061 8/16 bit processor @ 15 MHz
  - 8K bytes of ROM, 256 bytes of RAM
  - Controlled 8 vehicle functions
Old School (Ford Motor Co. EEC)
“Megasquirt”
Open Source Fuel Injection Controller

www.megasquirt.info
Automotive Trends
ECUs/vehicle, 2018

- Economy: 20+
- Volume Luxury: 50+
- Ultra Luxury: 100+

Communication via CANBus, Ethernet, wireless and other networks

After engine and transmission, the 3rd heaviest component in a modern automobile is the wiring harness.
"Turn hardware problems into software problems"...a lot of software problems.

- Lockheed Martin F-35 Lightning II: 30 million LOC

- Ford F-150, as per a claim made at the 2016 CES:
Complexity
Two F-Series Vehicles

- "Turn hardware problems into software problems"...a *lot* of software problems.
- Lockheed Martin F-35 Lightning II: 30 million LOC
- Ford F-150, as per a claim made at the 2016 CES: 150 million LOC
DJI NAZA N3
UAV Flight Controller

- 32-bit DSP, GPS, compass, gyro, power management, external LED driver, etc.
- Up to 8 rotors; street price $300
Engineering Challenges

Constraints typical in many ES designs

- **Form Factor**
  (size, shape, weight)

- **Environment**
  (shock, vibration, temperature, moisture, radiation, RF)

- **Power**
  (battery life, heat dissipation, surges)

- **Reliability, Security, Safety**
  Failures may cause injuries or property damage
  Mission critical but often inaccessible

- **Cost**
  $ (or €!)
Hardware Challenges

Form Factor

• Space Limitations
  Enclosure design often drives hardware design
  Highly-integrated chips minimize parts count
  (if you can turn off what you don’t need!)

• Close collaboration with Mechanical Engineers is critical to project success

• 3D Printing/Rapid Prototyping gets better every day, and can avoid costly mistakes later in production
Hardware Challenges

Form Factor

- Exotic PCB materials and techniques may be required
  Flexible circuits
  Very fine feature sizes
  Via-in-pad, laser-drilled microvias

- Tiny SMT devices, BGAs, etc. save space, but watch ratings, manufacturability and rework issues

  (01005 is really small, and it's tough to solder that kludge wire dead-center under a BGA package...)
Exotic PCB materials and techniques again
  High-temperature/pressure substrates
    (there's a reason FR-4 is cheap)
  Conformal coating
    (moisture, Pb-free Sn whiskers)

"Shake and Bake"
  Shock Mounting
  Ingress Prevention (IP-xx standards)
Electromagnetic Compatibility (EMC)

Emission:
Your device makes other devices not work

Susceptibility:
Other devices make your device not work

- Both are bad
- Requires careful attention to electronic and mechanical design to ensure compliance
Minimize power consumption for...
Battery life (per-charge and lifetime)
Heat Dissipation
Operating Cost

Designing for Power Management
(Not just a 5V rail and ground!)
Many switchable power domains
Integrated current and voltage monitoring
Voltage and clock speed scaling
Hardware Challenges

Power

- Not just "off" and "on"
  Low-power sleep modes
  Wake-up receivers, Bluetooth Low Energy, etc.
  Challenge: naming all the power modes!

- Beware of battery-draining "sneak paths"
  "Disabled" or "shut down" doesn't mean "off"
  Check those data sheets!

- External power has its own problems; design for and test extreme cases (surges, undervoltage, cranking).
For critical applications, redundant systems with a "computer third party" to disable failing units

- Power-On Self Test (POST) capabilities, self-diagnostics, system health reporting

- Predict how systems may fail (risk assessment, fault tree analysis, etc.)

- Minimize effects ("fail in the right direction")
Hardware Challenges

Reliability/Security/Safety

- Recover from software failures
  "watchdog" timers to restart after a crash

- Authentication and/or encryption to prevent or detect tampering with soft components

- Wireless or Internet-connected systems require added attention to security issues

- Audit and manage supply/manufacturing chain to ensure authenticity and quality of components
"To define it rudely but not inaptly, engineering is the art of doing that well with one dollar, which any bungler can do with two after a fashion."

--Arthur Mellen Wellington, c. 1887
Hardware Architecture

Embedded Processors

• Architecture Options for Every Design
  8-, 16-, 32-, 64-bit
  Optional floating-point, memory management

• High Integration
  Dedicated on-chip peripherals and coprocessors
  Minimizing chip count speeds design, saves power

• Complex, powerful power management
  Separate core, I/O, peripheral power
  Multiple low-power sleep/standby modes
  Companion PMICs for higher-end devices
Sophisticated Timing
- Accurate timing for realtime performance
- Fast wake-up time
- Very low power timers
- Watchdog timers to recover after failures

Pin Multiplexing
- Driving pins takes a lot of a chip’s power budget
- Multiplex out test signals, unused chip functions
TI MSP430F2001

Block Diagram

16-bit embedded CPU with Flash, Comparator, 10 I/O Lines, Serial Port
Low Performance, Low Price ($0.50)
TI MSP430F2001

Form Factor

- 14-pin TSSOP or 16-pin PQFP, 0.25 cm²
- Software configurable pinout (up to 5 choices for some pins)
- JTAG support for in-circuit testability

(Dozens of other designs available)
TI MSP430F2001
Power Management

The following six operating modes can be configured by software:

- **Active mode AM;**
  - All clocks are active
- **Low-power mode 0 (LPM0);**
  - CPU is disabled
  - ACLK and SMCLK remain active. MCLK is disabled
- **Low-power mode 1 (LPM1);**
  - CPU is disabled
  - ACLK and SMCLK remain active. MCLK is disabled
  - DCO’s dc-generator is disabled if DCO not used in active mode
- **Low-power mode 2 (LPM2);**
  - CPU is disabled
  - MCLK and SMCLK are disabled
  - DCO’s dc-generator remains enabled
  - ACLK remains active
- **Low-power mode 3 (LPM3);**
  - CPU is disabled
  - MCLK and SMCLK are disabled
  - DCO’s dc-generator is disabled
  - ACLK remains active
- **Low-power mode 4 (LPM4);**
  - CPU is disabled
  - ACLK is disabled
  - MCLK and SMCLK are disabled
  - DCO’s dc-generator is disabled
  - Crystal oscillator is stopped

6 Power Modes

Externally triggered wake-up in < 1us

220uA active; 0.1-0.5uA standby
TI MSP430F2001
Software Environment

• 16-MHz MSP430 Processor Core
  1,280 bytes of flash, 128 bytes of RAM
  Small assembly or C programs
  “Bare metal” (no operating system)

  Typical applications:
  Utility metering
  Simple portable medical devices
  Sensor networks
Overview

Embedded CPU with ARM, DSP cores

“Kitchen Sink” onboard peripheral set
- video display interface and acceleration
camera interface
- high speed serial interfaces
- multiple USB interfaces
- optimized for smartphone handsets

JTAG interface for debugging

High Performance = High Price ($35)
TI OMAP DM3730 Block Diagram
TI OMAP DM3730

Form Factor

515-pin Ball Grid Array (BGA), 1 cm

347 pins on bottom, 168 pins on top for P-O-P memory chip

Software configurable pinout (up to 8 choices for some pins)
TI OMAP DM3730
Power Management

9 Top Level Power Domains with independent supplies

PRCM (Power, Reset and Clock Module) “brain stem” can turn domains on and off

“SmartReflex” power and clock speed scaling

Up to 2W fully active; a few mW in standby modes

Companion power management chip (TPS65960) contains multiple power regulators, battery charger support, etc.
TI OMAP DM3730
Software Environment

- ARM Cortex A8 Processor
  - Full memory management
  - Linux main-line kernel tree support
  - Supervisory and User Interface functions

- TMS320C64x DSP Core
  - Access to private and system memory and I/O
  - Supervised by ARM, but autonomous
  - Dedicated DSP functions

- Task: Maximize efficiency by intelligent allocation of functions among ARM and DSP
Atmel ATtiny4
8-bit CPU in SOT-23

- Atmel ATtiny4
  6 pins, 512b flash, $0.44
  200μA @ 1.8V (active)
  25μA @ 1.8V (idle)
  0.1μA @ 1.8V (power-down)

Figure 1-1. Pinout of ATtiny4/5/9/10

SOT-23

(PCINT0/TPIDATA/OC0A/ADC0/AIN0) PB0
GND
(PCINT1/TPICLK/CLKI/ICP0/OC0B/ADC1/AIN1) PB1

1 6 PB3 (RESET/PCINT3/ADC3)
2 5 VCC
3 4 PB2 (T0/CLKO/PCINT2/INT0/ADC2)
FPGA Technology

- Field Programmable Gate Array
- “Grab Bag” of hardware on a single chip
- Configures itself on power-up, usually from a specialized low-cost serial FLASH memory
- Often used as a “front end” to reduce CPU processing demand or pin count

Xilinx (www.xilinx.com)
Intel, ex-Altera (www.intel.com)
FPGA Technology (continued)

- Powerful design tools generate FPGA “code” using software-like descriptions in Hardware Description Languages (VHDL, Verilog)
- Designs can be extensively simulated
- Designs can be converted into ASICs (Application Specific Integrated Circuit) for lower cost in very high volume products
FPGA Technology (continued)

- Many hardware functions can be purchased as IP (intellectual property) “Cores”—FPGA code that can be securely dropped into an existing FPGA design.

  (Audio/video codecs, network communication, radio modulation/demodulation, encryption/decryption...)

- The third-party marketplace for FPGA intellectual property has created a means to monetize design innovations without having to build hardware
A “Soft Core” CPU and peripherals can be implemented on an FPGA

Some CPU architectures are available as IP (intellectual property)--VHDL or Verilog code

IP also available for complicated I/O tasks (USB, FireWire, Ethernet, audio and video encoding/decoding, memory controllers, etc.)

True single-chip solution for some systems
Xilinx MicroBlaze™
Soft-Core Processor

- Implemented on a Xilinx FPGA
- Any combination of standard, custom or user-modified peripherals
- Shares FPGA with other user-defined hardware

Intel’s Nios II is comparable
Xilinx MicroBlaze™
On Spartan-3E FPGA

- Spartan XC3S1600E (~$50)
  376-pin BGA package (23mm x 23mm)
  33,192 “Logic Cells” (4-input LUTs)
  36 dedicated multipliers
  36 4K-byte RAM blocks
- MicroBlaze core itself uses ~1,000 LUTs
- Peripherals use additional space
  Parallel port (23), Ethernet (1,800)
- Typical instance is ~25% of a 1600E, 75% remaining for user hardware
Xilinx MicroBlaze™
On Spartan-6 FPGA

- Spartan-6 LX and LXT Series
  144 to 900 pins (8x8mm to 31x31 mm)
  3,800 to 147,000 Logic Cells (LUTs)
  $10-$320 (Q1K)

- Some units have PCI Express endpoints, multi-gigabit SerDes, DDR3 controller)

- Footprint compatibility for scalability
  (same board layout, different chips)
Microsemi Igloo Nano
Ultra low-power FPGA

• Range of form factors
  36 to 100 pins (3x3 mm to 14x14 mm)
  260 to 6,144 “VersaTiles” (D flip-flops)
  $4-$15

• Self-configuring from onboard Flash

• As low as 1.2V power supply

• 2 uW standby power in “freeze mode”

*formerly Actel
Microsemi Igloo Nano
Ultra low-power FPGA

- No-cost, Royalty-Free Processor Cores
  
  CoreABC: Very simple, programmed in assembly language, internal memory
  
  Core8051, CoreM1: More complex, programmed in C, external memory
  
  APB Bus: Lots of peripherals (UART, SPI, I2C, GPIO, Timers, 10/100 Ethernet, etc.)
Microsemi Igloo Nano Starter Kit ($99)
Xilinx Zynq-7000
Hybrid FPGA

- "PS" (Processor Subsystem): 2 ARM Cortex-A9 processors and a lot of useful peripherals
- "PL" (Programmable Logic): Configurable FPGA logic blocks (multiple sizes available)
- Standard buses connecting PS and PL; interrupts
- Some I/O pins accessible directly by PS, so can be used prior to loading FPGA bitstream
- Best of both worlds (for some designs)
Xilinx Zynq-7000
Hybrid FPGA
Embedded systems are found in almost all product areas due to the power of being able to combine hardware and software in the most appropriate way to solve a given problem.

ES present significant engineering challenges in terms of physical design, environment, power, reliability, security, safety and cost.

Developing embedded systems requires balancing many competing factors.

Many specialized processors and support chips are optimized specifically for embedded applications.
Part 2 Preview
(Software)

- ARM and DSP Architectures
- Software challenges in Embedded Systems
- Key decisions in ES software development
- ARM and DSP Architectures
- Low-cost ES Prototyping Platforms
- Trends and opportunities in the ES industry
Questions

alix@ieee.org