Microcontroller & Embedded Systems-Module-3-2

Dr. Girijamma H A
Professor
Department of Computer Science and Engineering
RNS Institute of Technology
Bangalore -560098
Email: girijakasal@gmail.com
Contact: 9480031494

Module-3-2-Contents

ESTD : 2001
An Institute with a Difference

2. The Typical Embedded System

3.2.1Core of an Embedded System

3.2.2 Memory

3.2.3 Sensors and Actuators

3.2.4 Communication Interface

3.2.5 Embedded firmware

3.2.6 Other system components.

Text book 2: Shibu K V, “Introduction to Embedded Systems”, Tata McGraw Hill Education, Private Limited, 2nd Edition.
Chapter 1(Sections 1.2 to 1.6) ,Chapter 2(Sections 2.1 to 2.6)

The Typical Embedded System Components

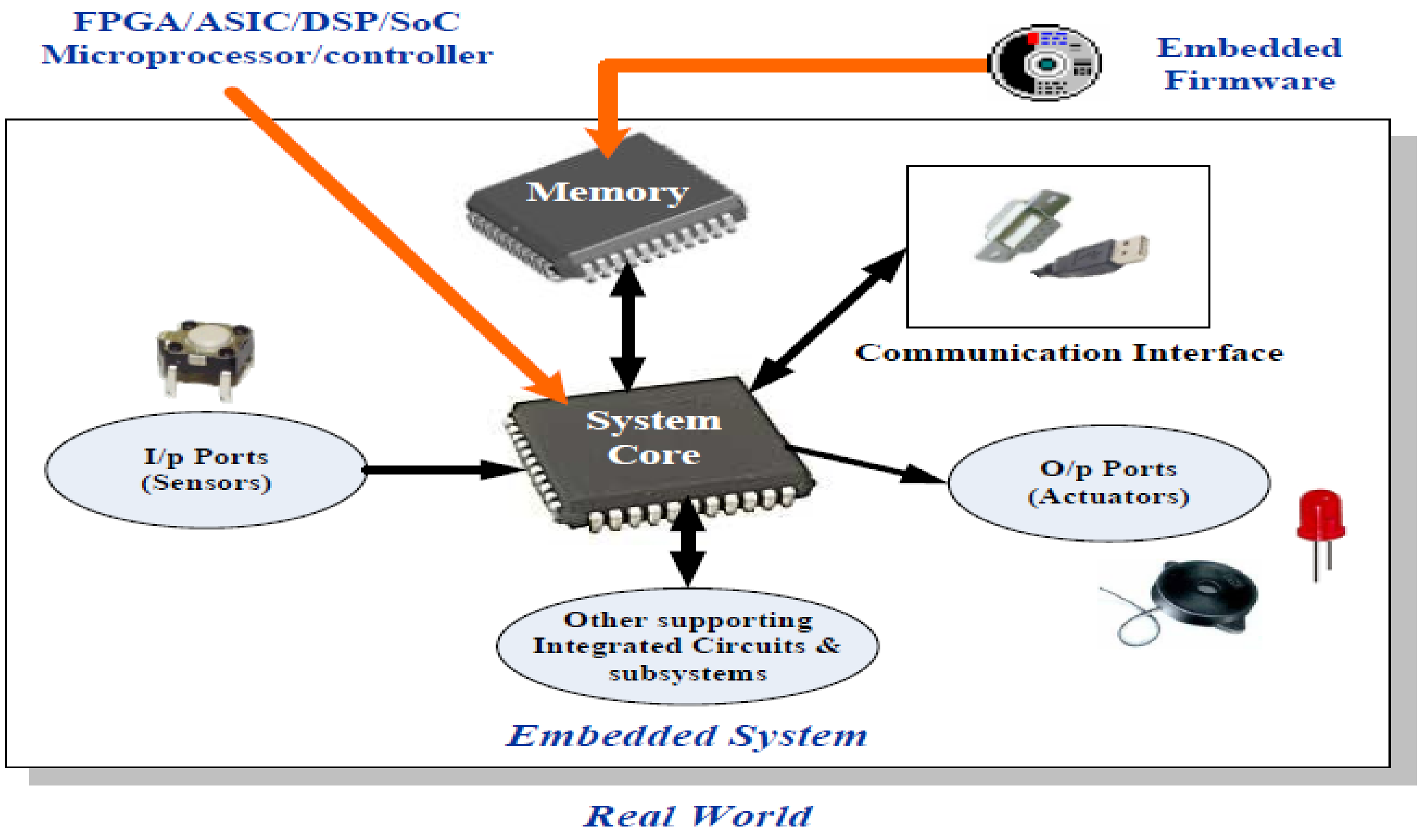


Fig: Elements of an Embedded System

- System Core
- Sensors
- Actuators
- Memory
- Other Supporting ICs

3.2.1 Core of an Embedded System

- Embedded systems are domain and application specific and are built around a central core.
- The core of the embedded system falls into any one of the following categories:
- 3.2.1.1 General Purpose and Domain Specific Processor
  - Microprocessors
  - Microcontrollers
  - Digital Signal Processors
- 3.2.1.2 Application Specific Integrated Circuits (ASICs)
- 3.2.1.3 Programmable Logic Devices (PLDs)
- 3.2.1.4 Commercial off-the-shelf Components (COTS)

3.2.1.1 General Purpose and Domain Specific Processor

ESTD : 2001
An Institute with a Difference

- Almost 80% of the embedded systems are processor/controller based.
- The processor may be a microprocessor or a microcontroller or a digital signal processor, depending on the domain and application.
- Most of the embedded systems in the industrial control and monitoring applications make use of the commonly available microprocessors or microcontrollers
- whereas domains which require signal processing such as speech coding, speech recognition, etc. make use of special kind of digital signal processors supplied by manufacturers like, Analog Devices, Texas Instruments, etc.

Microprocessors

ESTD : 2001
An Institute with a Difference

- The CPU contains the Arithmetic and Logic Unit (ALU), Control Unit and Working registers
- Microprocessor is a dependent unit and it requires the combination of other hardware like Memory, Timer Unit, and Interrupt Controller etc for proper functioning.
- Intel claims the credit for developing the first Microprocessor unit Intel 4004, a 4 bit processor which was released in Nov 1971

GPP vs. Application-Specific Instruction Set Processor (ASIP)

- General Purpose Processor or GPP is a processor designed for general computational tasks
- GPPs are produced in large volumes and targeting the general market.
- ASIPs are processors with architecture and instruction set optimized to specific domain/application requirements
- Network processing, Automotive, Telecom, media applications, digital signal processing, control applications etc. are examples for ASIPs

Microcontrollers

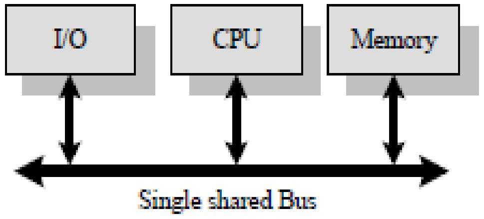
- A highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General -purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports.
- Microcontroller can be general purpose (like Intel 8051, designed for generic applications and domains) or application specific (automotive applications)
- Microcontrollers are cheap, cost effective and are readily available in the market
- Microprocessor vs Microcontroller

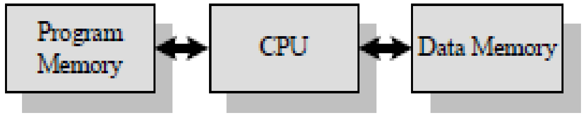
Digital Signal Processors (DSPs)

ESTD : 2001
An Institute with a Difference

- Powerful special purpose 8/16/32 bit microprocessors
- Are 2 to 3 times faster than the general-purpose microprocessors in signal processing applications
- DSP can be viewed as a microchip designed for performing high speed computational operations for 'addition', 'subtraction', 'multiplication' and 'division'
- A typical Digital Signal Processor incorporates the following key units
  - Program Memory , Data Memory, Computational Engine, I/O Unit
- RISC vs. CISC Processors/Controllers

Harvard V/s Von-Neumann Processor/Controller Architecture





Von-Neumann Architecture

Harvard Architecture

Harvard Architecture	Von-Neumann Architecture
Separate buses for instruction and data fetching	Single shared bus for instruction and data fetching
Easier to pipeline, so high performance can be achieved	Low performance compared to Harvard architecture
Comparatively high cost	Cheaper
No memory alignment problems	Allows self-modifying codes
Since data memory and program memory are stored physically in different locations, no chances for accidental corruption of program memory	Since data memory and program memory are stored physically in the same chip, chances for accidental corruption of program memory

Big-endian V/s Little-endian processors

- **Little-endian** means the **lower-order byte** of the data is stored in memory at the lowest address, and the **higher-order byte** at the highest address. (The little end comes first)
- **Big-endian** means the **higher-order** byte of the data is stored in memory at the lowest address, and the **lower-order** byte at the highest address. (The big end comes first.)

Little-endian Operation

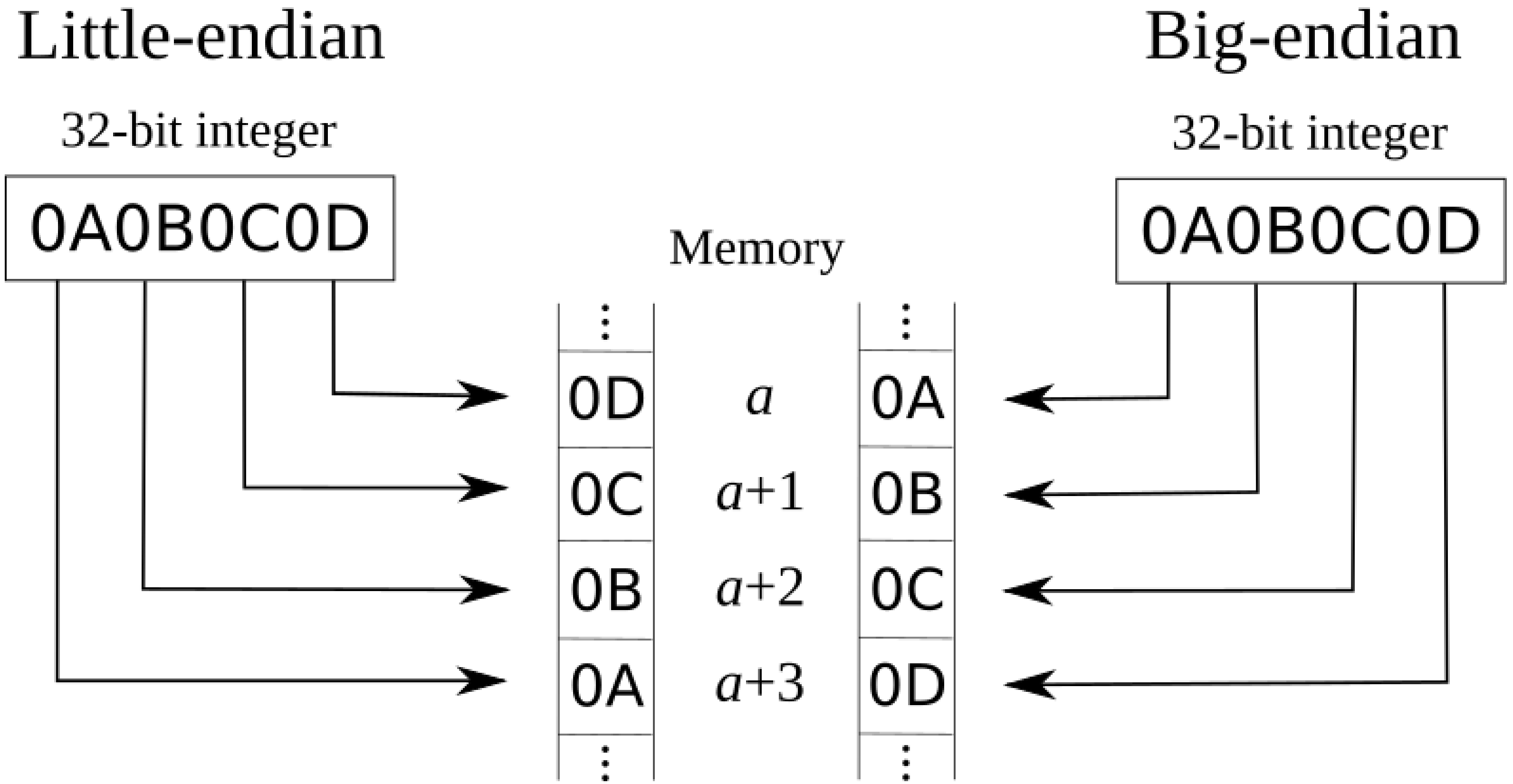
Base Address + 0	Byte 0	Byte 0	0x20000 (Base Address)
Base Address + 1	Byte 1	Byte 1	0x20001 (Base Address + 1)
Base Address + 2	Byte 2	Byte 2	0x20002 (Base Address + 2)
Base Address + 3	Byte 3	Byte 3	0x20003 (Base Address + 3)

Big-endian Operation

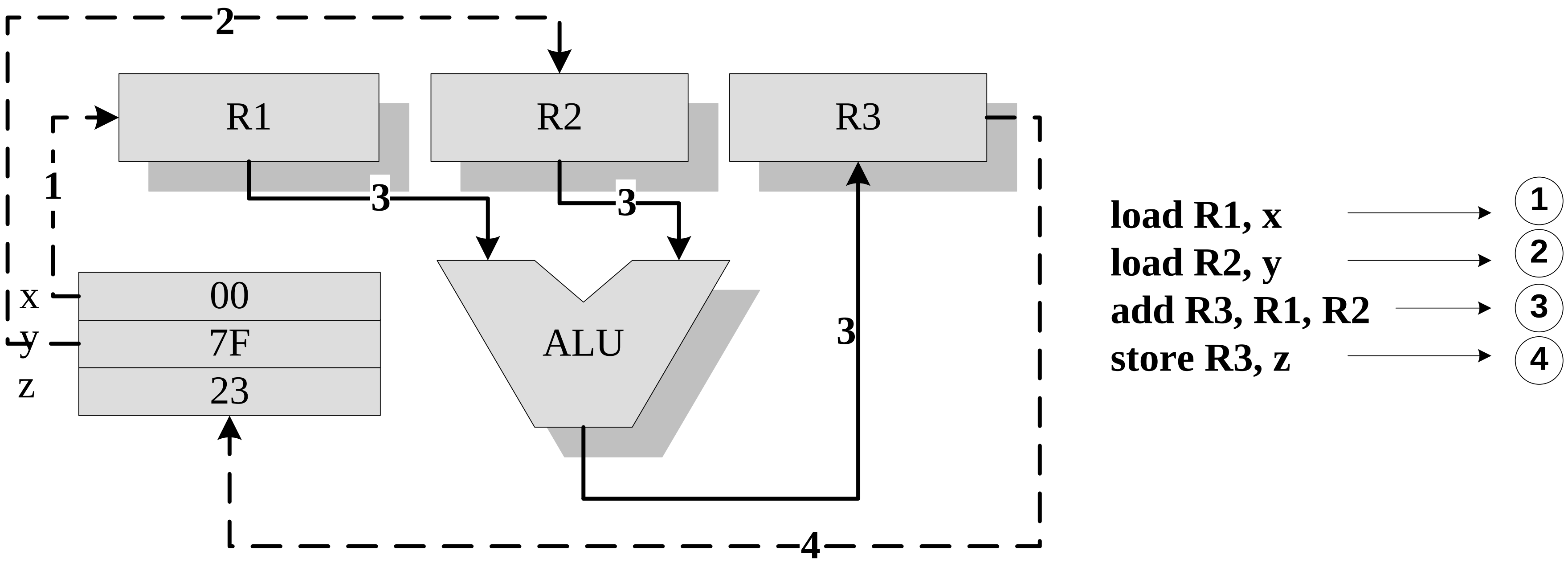
Base Address + 0	Byte 3	Byte 3	0x20000 (Base Address)
Base Address + 1	Byte 2	Byte 2	0x20001 (Base Address + 1)
Base Address + 2	Byte 1	Byte 1	0x20002 (Base Address + 2)
Base Address + 3	Byte 0	Byte 0	0x20003 (Base Address + 3)

Big-endian V/s Little-endian processors

➢



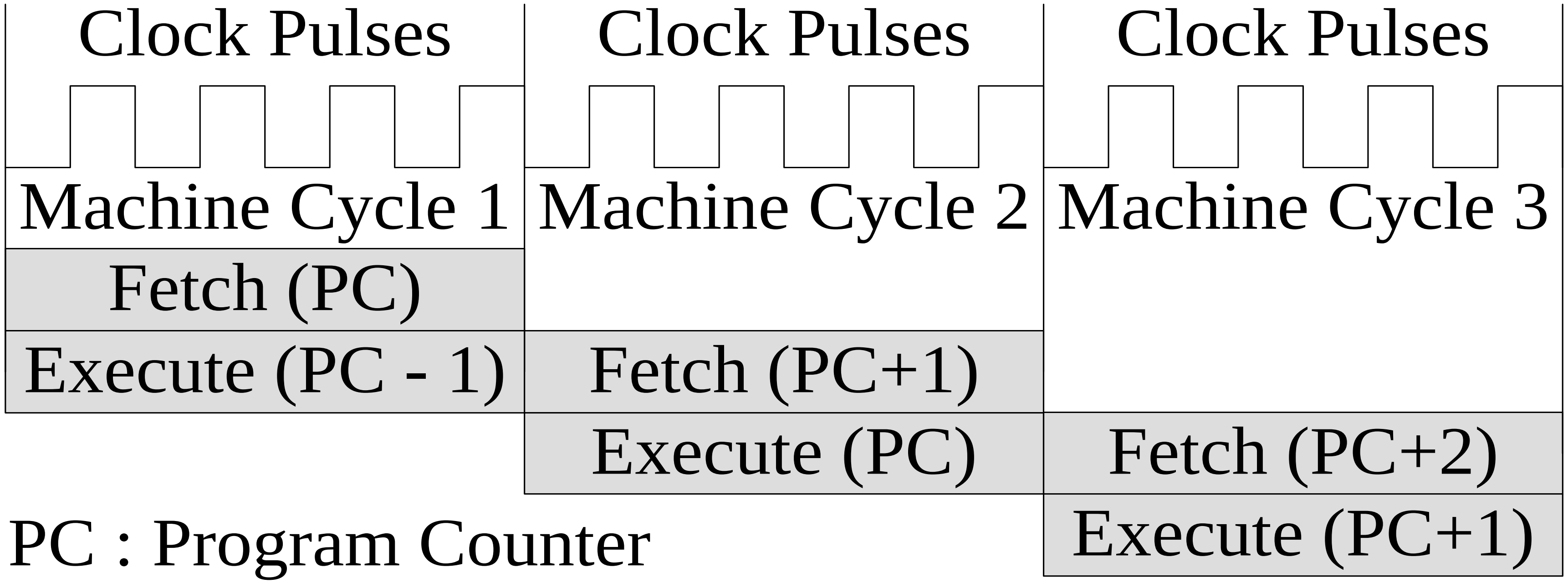
Load Store Operation & Instruction Pipelining



- Suppose x, y and z are memory locations and we want to add the contents of x and y and store the result in location 2. Under the load store architecture, the same is achieved with 4 instructions as shown in Figure.

Load Store Operation & Instruction Pipelining

ESTD : 2001
An Institute with a Difference



- Figure illustrates the concept of Instruction pipelining for single stage pipelining.
- For simplicity let’s consider decode and execution together.

3.2.1.2 Application Specific Integrated Circuits (ASICs)

- ASIC is a microchip designed to perform a specific or unique application
- It integrates several functions into a single chip
- As a single chip, ASIC consumes a very small area
- ASICs can be pre-fabricated for a special application
- ASIC based systems are profitable only for large volume commercial productions
- Fabrication of ASICs requires a non-refundable initial investment for the process technology and configuration expenses. This investment is known as Non-Recurring Engineering Charge (NRE) and it is a one-time investment.

3.2.1.3 Programmable Logic Devices (PLDs)

ESTD : 2001
An Institute with a Difference

- Logic devices-classification - **Fixed and Programmable**
- The circuits in a **fixed logic** device are permanent
- PLDs can be re-configured to perform any number of functions at any time
- Designers can use inexpensive software tools to quickly develop, simulate, and test their logic designs in PLD based design.
- The design can be quickly programmed into a device, and immediately tested in a live circuit
- Based on re-writable memory technology and the device is reprogrammed to change the design

Advantages of PLDs:

- PLDs offer customers much more flexibility during the design cycle
- PLDs do not require long lead times for prototypes or production parts
- PLDs do not require customers to pay for large NRE costs
- PLDs allow customers to order just the number of parts they need
- PLDs can be reprogrammed even after a piece of equipment is shipped to a customer
- the ability to add new features or upgrade products that already are in the field.
- they simply upload a new programming file to the PLD, via the Internet, creating new hardware logic in the system.

CPLDS and FPGAs

- The two major types of PLDs are
  - **Field Programmable Gate Arrays (FPGAS) and**
  - **Complex Programmable Logic Devices (CPLDS).**
- FPGAS offer the highest amount of logic density,

  the most features, and the

  highest performance.
- The largest FPGA now shipping, part of the Xilinx VirtexTM line of devices, provides eight million “system gates
- FPGAs are used in a wide variety of applications ranging from data processing and storage, to instrumentation, telecommunications, and digital signal processing.

Comparison between FPGA & CPLD

Sr. No.	FPGA	CPLD
1	It is field programmable gate arrays.	It is complex programmable logic device.
2	Capacity is defined in terms of number of gates available.	Capacity is defined in terms of number of macro-cells available.
3	FPGA consumes less power than CPLD	CPLD consumes more power than FPGA devices.
4	Numbers of input and output pins on FPGA are less than CPLD.	Numbers of input and output pins on CPLD are high.
5	FPGA is suitable for designs with large number of simple blocks with few numbers of inputs.	CPLD are ideal for complex blocks with large number of inputs.
6	FPGA based designs require more board space and layout complexity is more.	CPLD based designs need less board space and less board layout complexity.
7	It is difficult to predict the speed performance of design.	It is easier to predict speed performance of design.
8.	FPGA are available in wide density range.	CPLDs contain fewer registers but have better performance.

3.2.1.4 Commercial off- the- Shelf Component (COTS)

- A Commercial off-the-shelf (COTS) product is one which is used ‘as-is’
- COTS products are designed in such a way to provide easy integration and interoperability with existing system components
- A COTS component in turn contains a General Purpose Processor (GPP) or Application Specific Instruction Set Processor (ASIP) or Application Specific Integrated Chip (ASIC)/Application Specific Standard Product (ASSP) or Programmable Logic Device (PLD)
- The major advantage of using COTS is that they are readily available in the market, cheap and a developer can cut down his/her development time to a great extent.

COTS-cont

- Examples for the COTS hardware unit are
  - Remote Controlled Toy Car control unit including the RF Circuitry part,
  - High performance, high frequency microwave electronics (2 to 200 GHz),
  - High bandwidth analog-to-digital converters,
  - Devices and components for operation at very high temperatures,
  - Electro-optic IR imaging arrays,
  - UV/IR Detectors etc

Thank You