Microprocessor & Computer Architecture (µpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

Microprocessor & Computer Architecture (µpCA)

Introduction to ARM

Dr. D. C. Kiran
Department of Computer Science and Engineering

Microprocessor & Computer Architecture (µpCA)

Syllabus

Unit 1: Basic Processor Architecture and Design

- ~~Microprocessor Overview~~
- ~~CISC VS RISC~~
- Introduction to ARM Processor & Applications
- ARM Architecture Overview
- Different ARM processor Modes

Unit 2: Pipelined Processor and Design

Unit 3: Memory Design

Unit 4: Input/Output Device Design

Unit 5: Advanced Architecture

ARM: [ACORN RISC MACHINE]

- Leading provider of 32-bit embedded RISC microprocessors, 75% of market
  - High performance
  - Low power consumption
  - Low system cost
- Solutions for
  - Embedded real-time systems for mass storage, automotive, industrial and networking applications
  - Secure applications - smartcards and SIMs
  - Open platforms running complex operating systems

Microprocessor & Computer Architecture (μpCA)

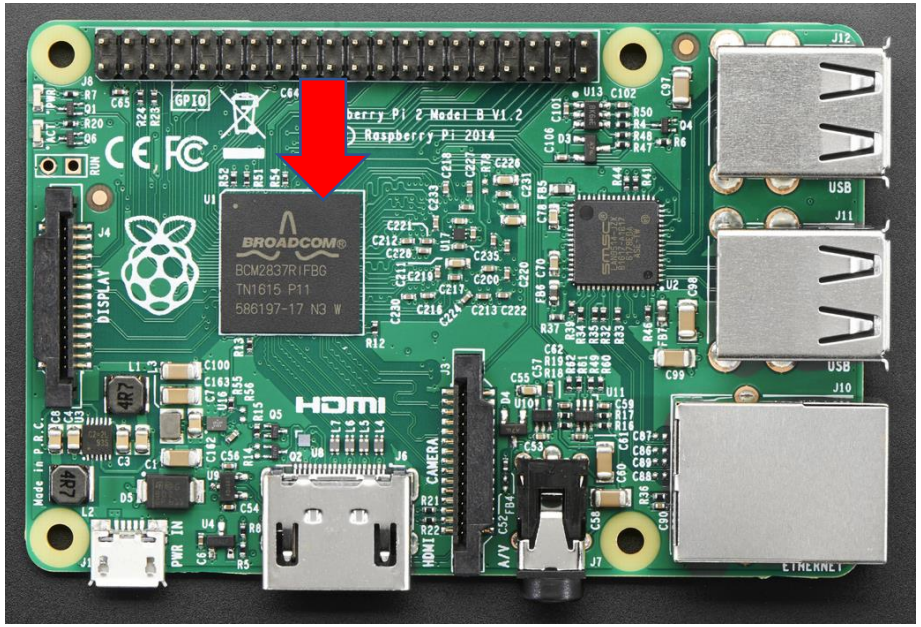
ARM Processor

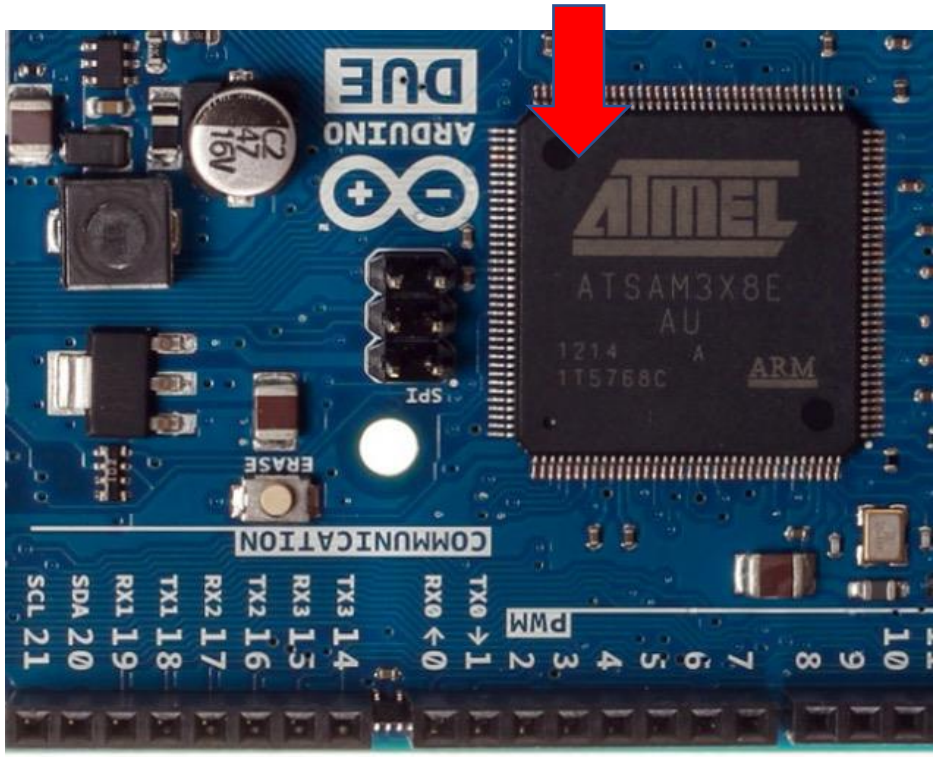
Microprocessor & Computer Architecture (µpCA)

Microprocessor & Computer Architecture (µpCA)

1: The **Raspberry Pi** 4 1.5 GHz 64-bit quad-core **ARM** Cortex-A72 (BCM2837)processor.

2: The **Arduino Due** : Atmel SAM3X8E ARM Cortex-M3 CPU.





Microprocessor & Computer Architecture (µpCA)

- ARM Based Mine Detection Robot Using GPS Technology
- ARM Based Hospital Enquiry System
- Temperature and Humidity Control System Using ARM and Graphical LCD
- Multi Functional Car With Accident Alert Sensors Using ARM
- ARM Based Vehicle Tracking System Using GPS and GSM
- Biometric fingerprint Identification based Bank Locker Security System Using ARM
- ARM Base Automated Bus Arrival Announcement System for the Blind Persons.
- ARM Based Digital Notice Board Using GSM

The ARM Architecture Versions

- **Version 1 (ARM1):** 26 bit addressing, no coprocessor.
- **Version 2 (ARM2):** Includes a 32 bit result multiply coprocessor
- **Version 2as (ARM3 & 250):**
- **Version 3 (ARM6 ,7,8):** 32 bit addressing
- **Version 4 (Strong ARM, ARM9):** Half word load/store instructions were provided.
- **Version 4T:** Thumbing: 16 bit instructions can be compressed in a 32 bit processor, thus enabling more instructions to be packed in the same memory, thereby increasing the code density.
- **Version 5T and 5TE (ARM10):** 5TE: thumb extension- built for powerful computations.
- CORTEX-M, CORTEX-R, CORTEX-A (32 Bit and 64 bit), NEOVERSE

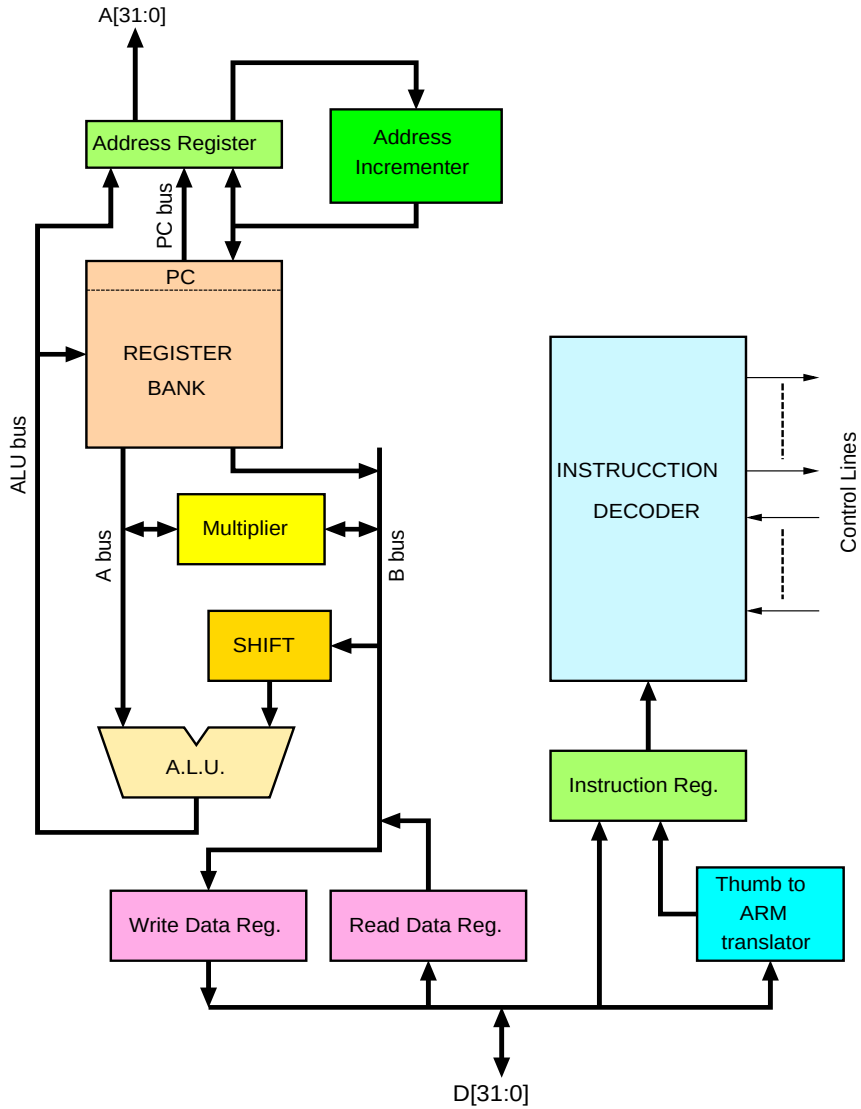
Reference
https://www.cs.umd.edu/~meesh/cmsc411/website/proj01/arm/armchip.html

ARM7TDMI Processor

- Current low-end ARM core for applications like digital mobile phones
- TDMI
  - **T**: Thumb, 16-bit instruction set
  - **D**: on-chip Debug support, enabling the processor to halt in response to a debug request
  - **M**: enhanced Multiplier, yield a full 64-bit result, high performance
  - **I**: EmbeddedICE hardware
- Von Neumann architecture
- 3-stage pipeline

Microprocessor & Computer Architecture (µpCA)

- Von Neumann Architecture
- 3-Stage Pipeline
  - Fetch, Decode, Execute
- 32-bit Data bus
- 32-bit Address Bus
- 37 32-bit Registers
- 32-bit ARM Instruction Set
- 16-bit THUMB instruction Set
- 32x8 Multiplier
- Barrel Shifter

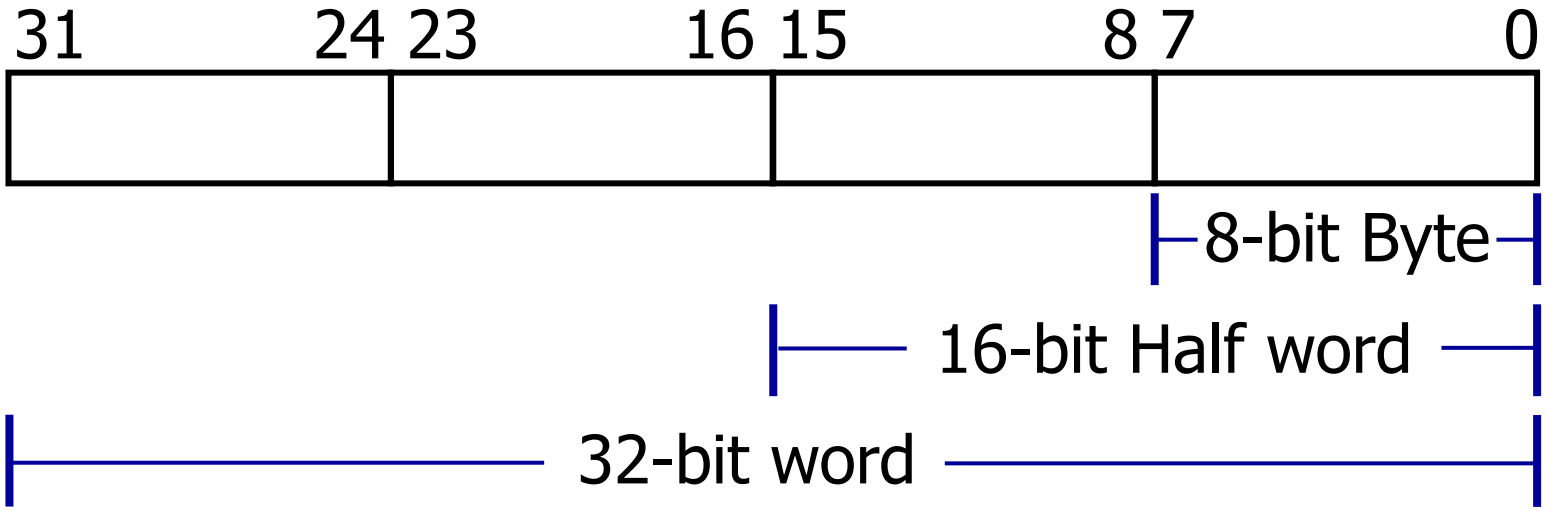


Data Sizes and Instruction Sets

- The ARM is a 32-bit architecture.

- When used in relation to the ARM:
  - **Byte** means 8 bits
  - **Halfword** means 16 bits (two bytes)
  - **Word** means 32 bits (four bytes)

- Most ARM's implement two instruction sets
  - 32-bit ARM Instruction Set
  - 16-bit Thumb Instruction Set



ARM processors support six data types.

- **8-bit** signed and unsigned bytes.
- **16-bit** signed and unsigned half-words; these are aligned on 2-byte boundaries.
- **32-bit** signed and unsigned words; these are aligned on 4-byte boundaries.

Processor Modes

The ARM has seven basic operating modes:

User

FIQ (High Priority)

IRQ (Low Priority)

Supervisor

Abort

Undefined

System

Modes other than User mode are collectively known as Privileged modes. Privileged modes are used to service interrupts or exceptions, or to access protected resources.

What happens when Program is Executing?

Program may Complete its execution

Or

Execution may be halted Temporarily or Permanently

Why Program Execution will be halted Temporarily?

Function / Subroutine Call (User Program)

Or

Operating System want to perform few maintenance work

Or

Error Handling Operation to be Executed, (Also called Interrupt Handling)

User Program will have lowest priority

Context switch with System Program and User Program

System Programs may have same or different priority

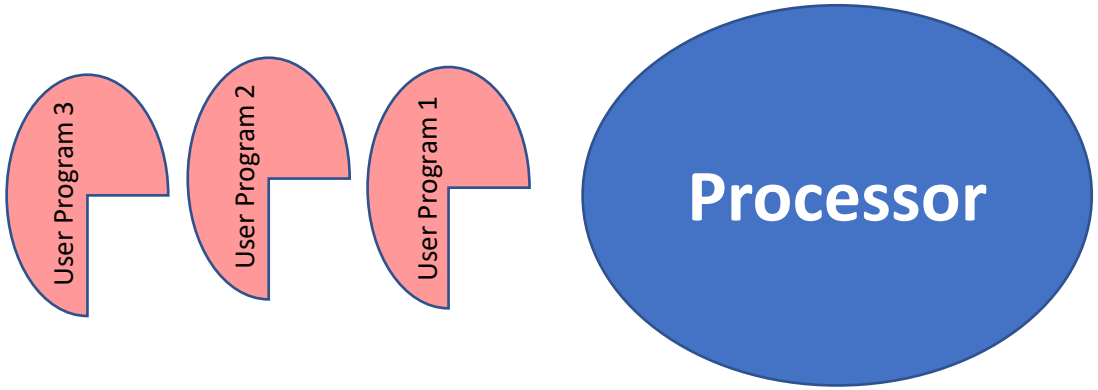
If Higher Priority program interrupts, Lower Priority Program will be switched.

If same priority program interrupts, no context switch take place.

During Context switching data related to program should be saved

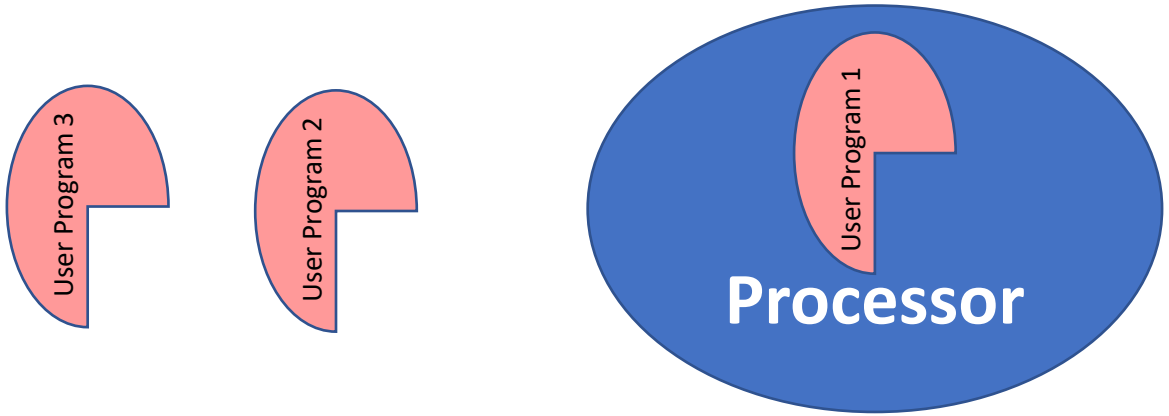
During Return, saved date should be restored back to program.

Processor Modes



User Program 1, User Program 2 and User Program 3 are waiting for CPU time for execution

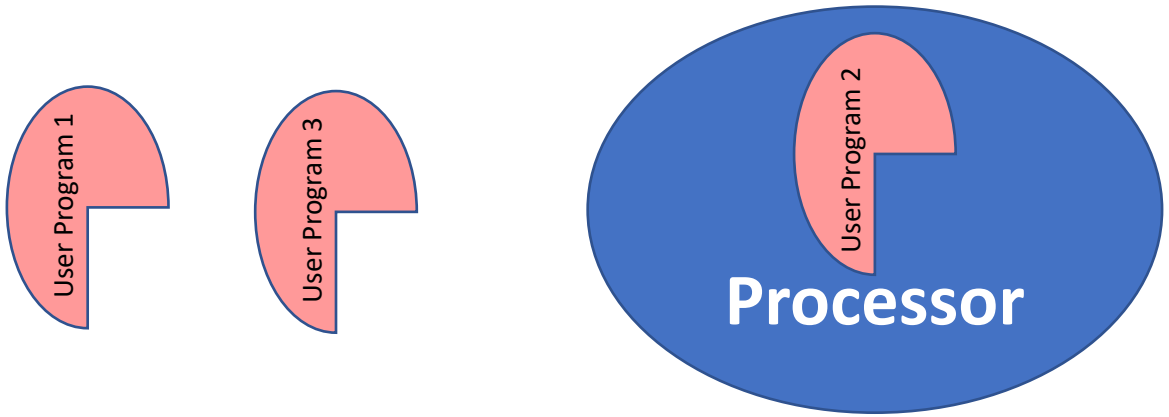
Processor Modes



User Program 1 is Executing

User Program 2 and User Program 3 are waiting for CPU time for execution

Processor Modes

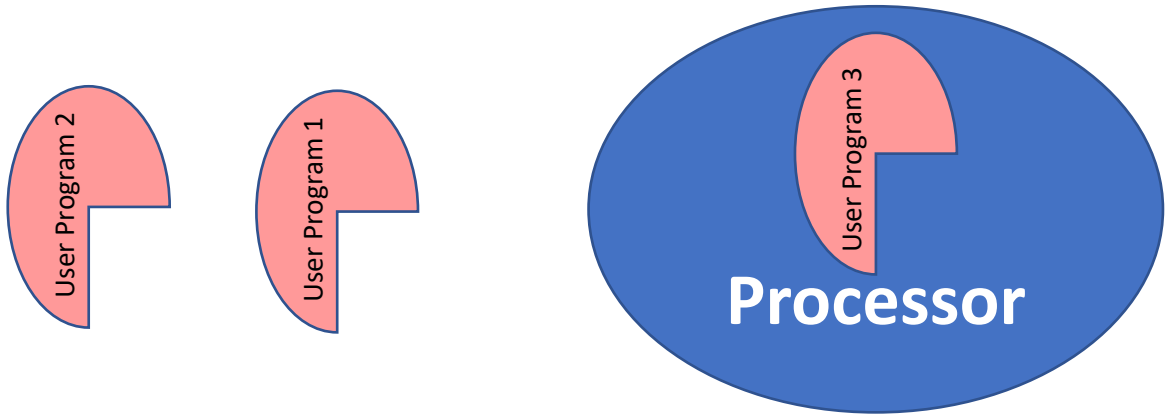


Context Switching between User Program 1 and User Program 2

User Program 2 is Executing

User Program 3 and User Program 1 are waiting for CPU time for execution

Processor Modes

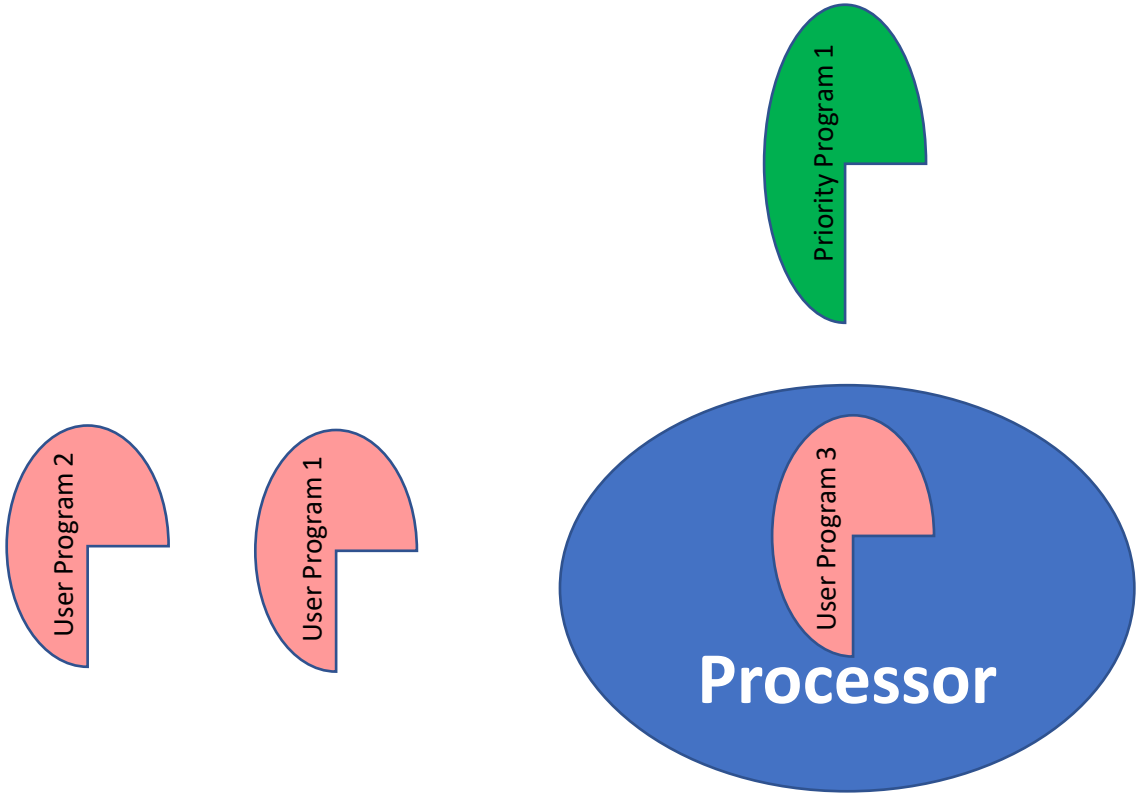


Context Switching between User Program 2 and User Program 3

User Program 3 is Executing

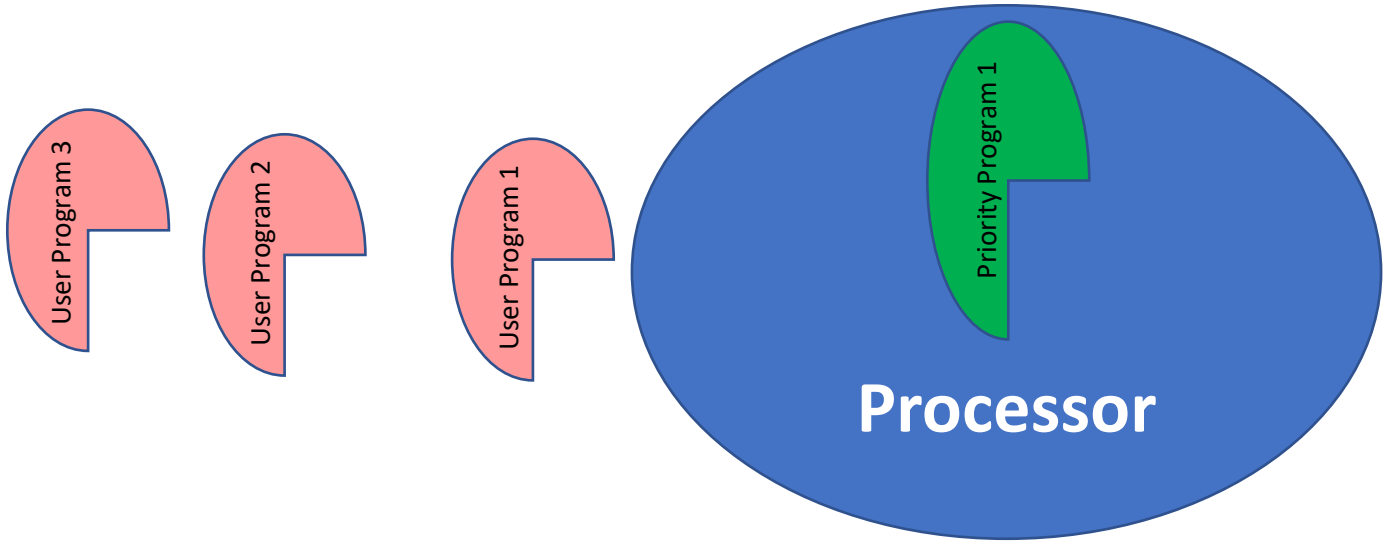
User Program 1 and User Program 2 are waiting for CPU time for execution

Processor Modes



Priority Program Arrive when User Program 3 is executing.
User Program 1 and User Program 2 are waiting for CPU time for execution

Processor Modes

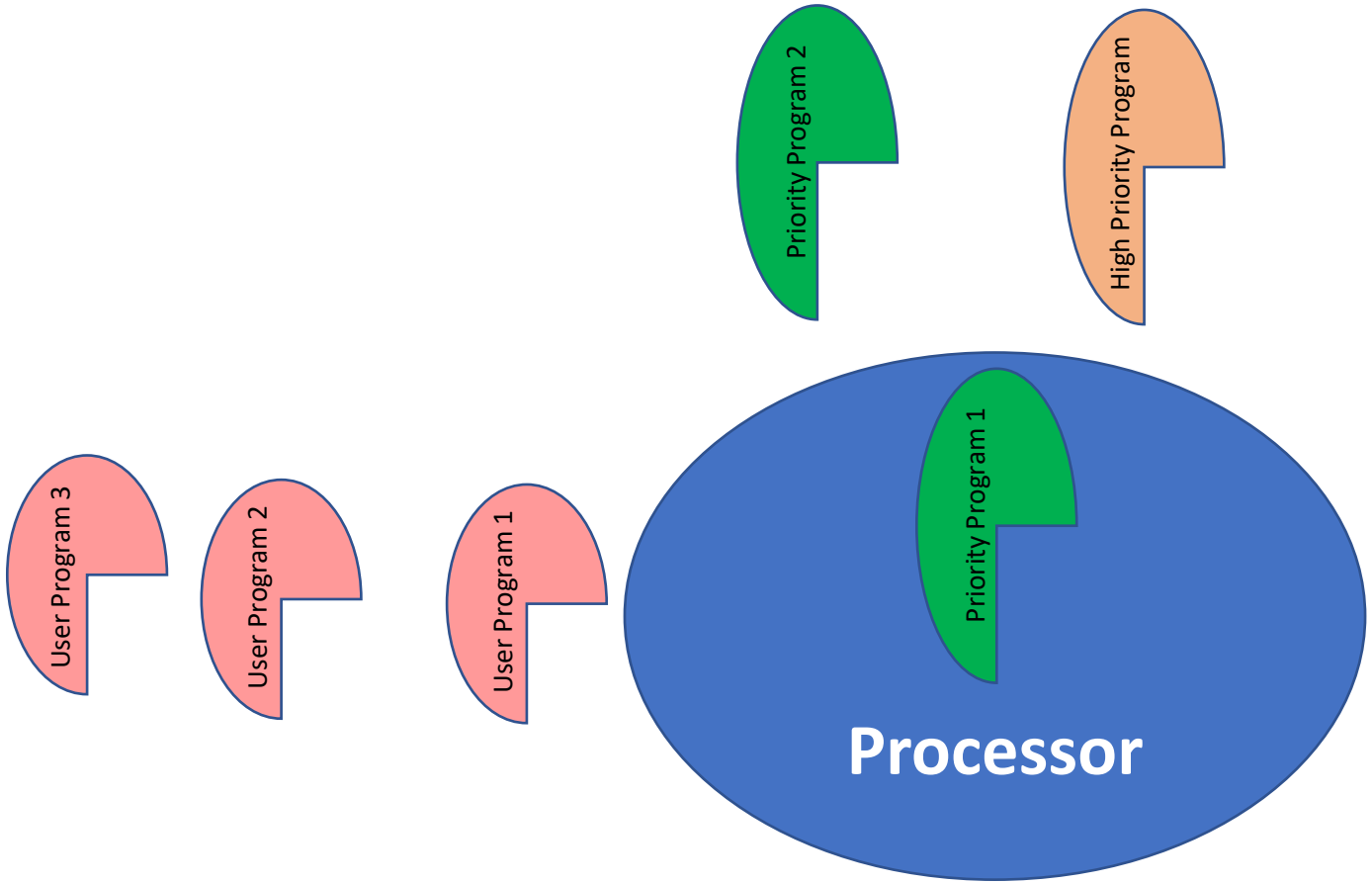


Context Switching between User Program 3 and Priority Program

Priority Program is Executing

User Program 1, 2 and 3 are waiting for CPU time for execution

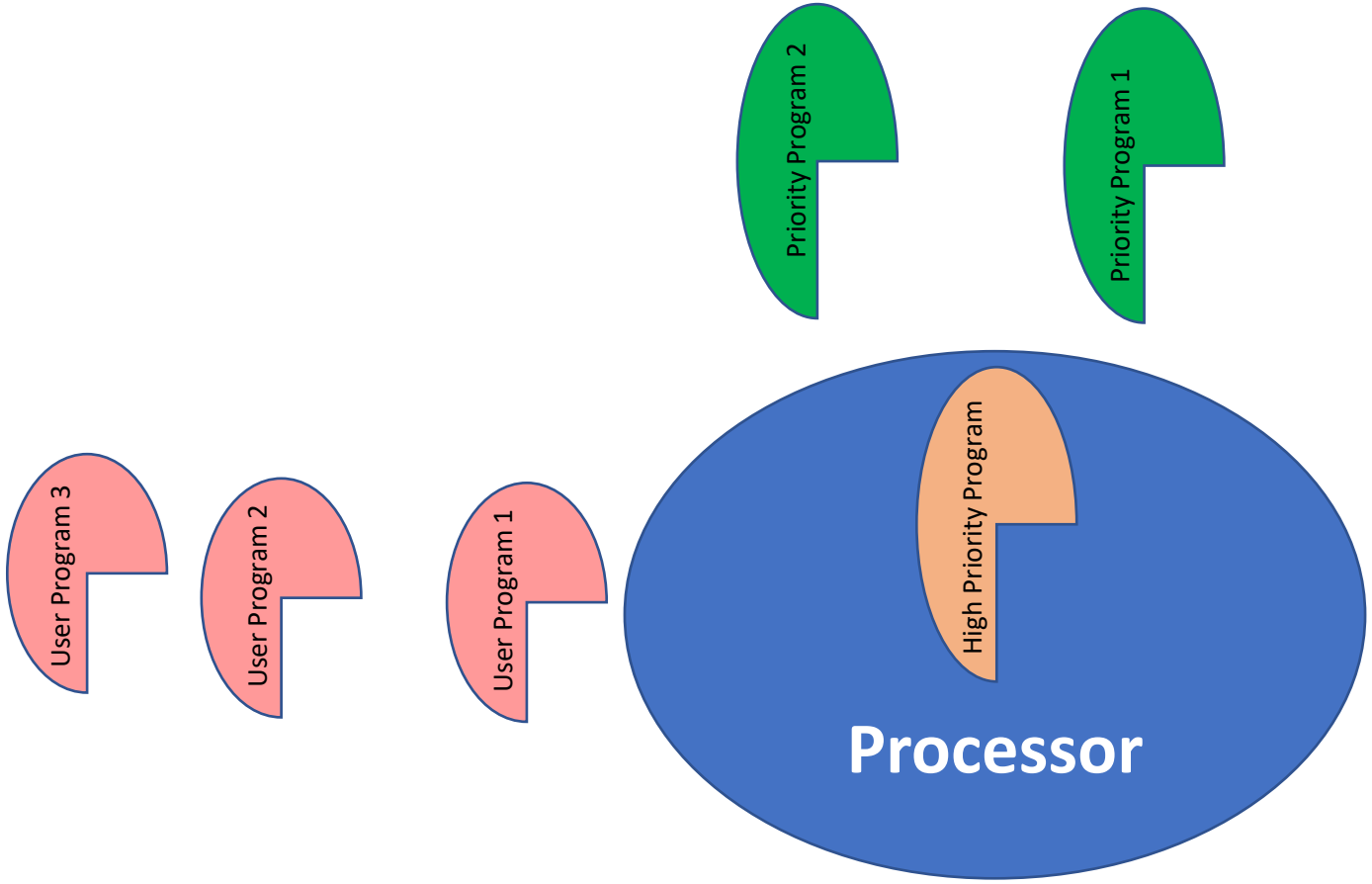
Processor Modes



High Priority Program and a Low Priority Program Arrive, when Priority program is Executing

User Program 1, 2 and 3 are waiting for CPU time for execution

Processor Modes



High Priority Program and a Low Priority Program Arrive, when Priority program is Executing

Context Switching between High Priority Program and Priority Program

High Priority Program is Executing

User Program 1, 2 and 3 are waiting for CPU time for execution

Processor Modes

The ARM has seven basic operating modes:

- **User** : Unprivileged mode under which most tasks run
- **FIQ** : Entered when a high priority (fast) interrupt is raised
- **IRQ** : Entered when a low priority (normal) interrupt is raised
- **Supervisor** : Entered on reset and when a Software Interrupt instruction is executed
- **Abort** : Used to handle memory access violations
- **Undef** : Used to handle undefined instructions
- **System** : Privileged mode using the same registers as user mode

Modes other than User mode are collectively known as Privileged modes. Privileged modes are used to service interrupts or exceptions, or to access protected resources.

Microprocessor & Computer Architecture (µpCA)

Next Class

- ~~Microprocessor Overview~~
- ~~CISC VS RISC~~
- ~~Introduction to ARM Processor & Applications~~
- ~~ARM Architecture Overview~~
- ~~Different ARM processor Modes~~
- Register Bank

THANK YOU

Dr. D. C. Kiran
Department of Computer Science and Engineering

dckiran@pes.edu

9829935135