BSC~Microsoft Research Centre

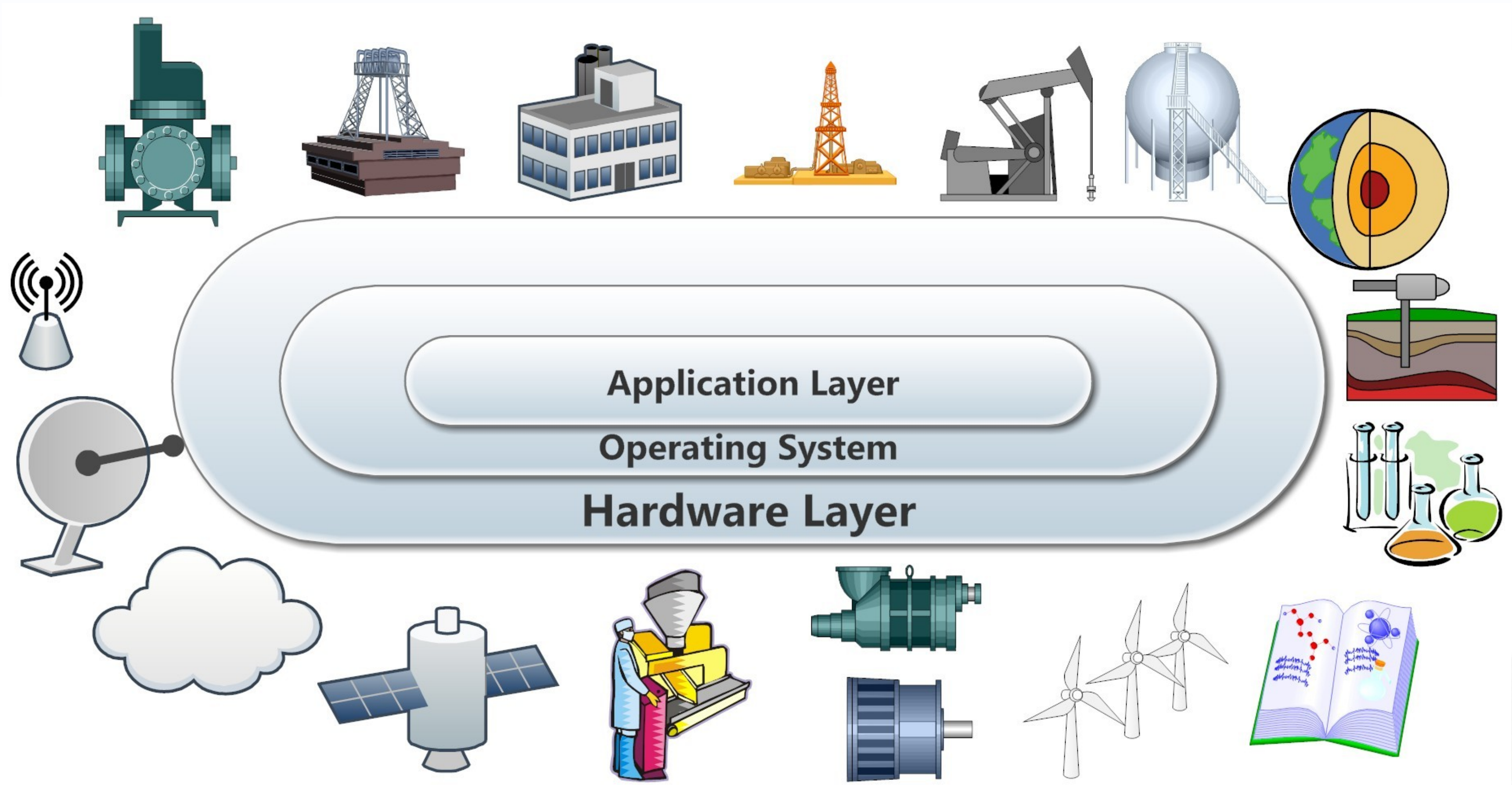
Computer Architecture

Tassadaq Hussain
Riphah International University
Islamabad Pakistan
Microsoft Barcelona Supercomputing Center
Universitat Politécnica de Catalunya
Barcelona, Spain

RIPHAH INTERNATIONAL UNIVERSITY

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

A Generic System



RIPHAH INTERNATIONAL UNIVERSITY

Hardware Architecture

- Processors
- Buses
- Memories
- Peripherals

Basic parameters of processor

Clock Cycle

Instruction Set

Pipeline

Registers

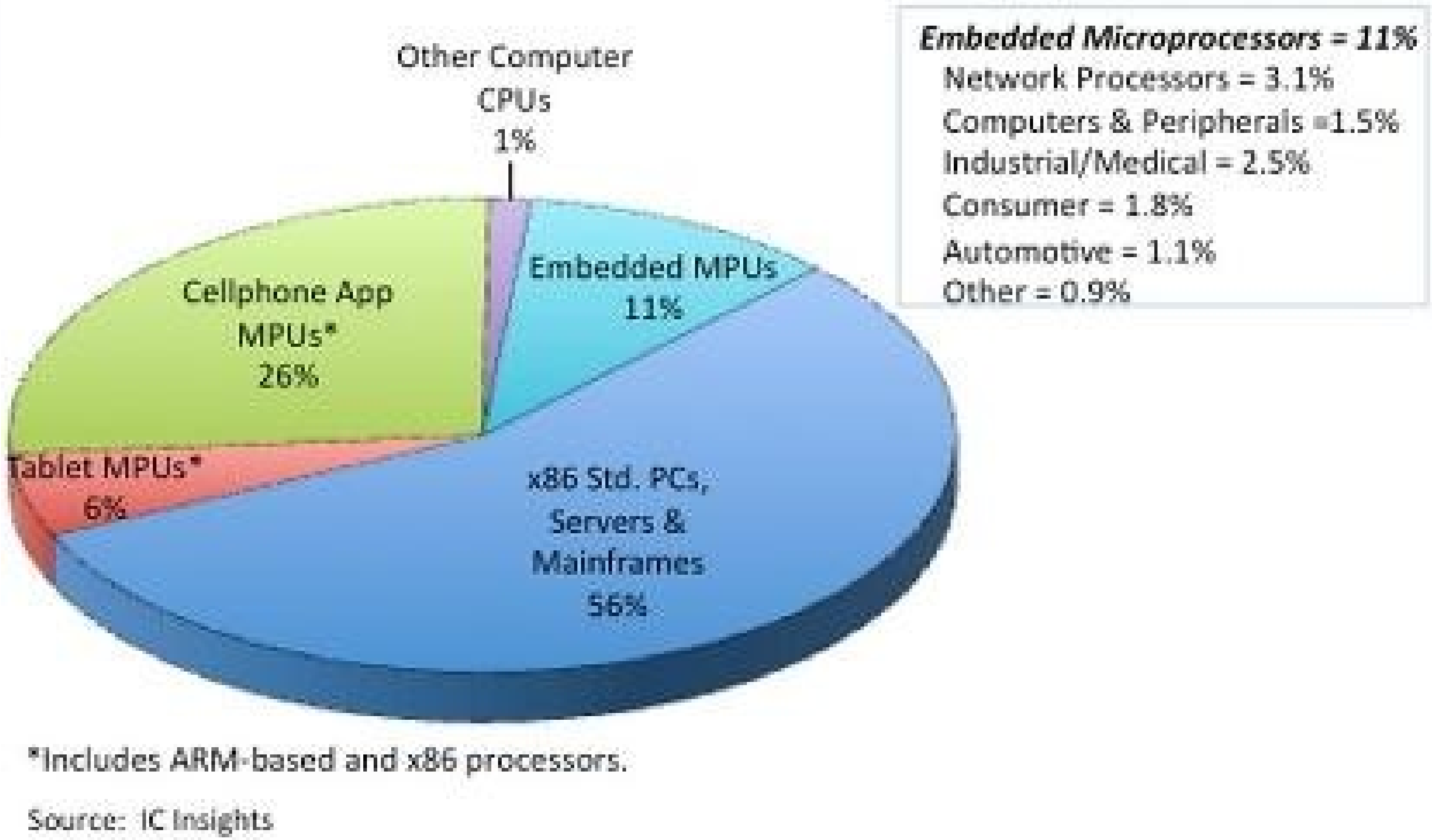
Embedded Processor Architectures

ISA: Instruction Set Architecture

- ARM: Low Power Low Cost
- x86: High Performance

- High-Performance x86 and ARM
- Industry-Leading & Most Efficient GPUs[1,2]
- Scalable Designs
- Memory Innovation
- Open Approach

Processor Market



RIPHAH INTERNATIONAL UNIVERSITY

Leading MPU Suppliers ($M)

2012 Rank	Company	2011	2012	Percent Change	Percent Marketshare	Main Product Lines
1	Intel	37,435	36,892	-1%	65.3%	x86 PC, server MPUs
2	Qualcomm	4,152	5,322	28%	9.4%	ARM mobile app processors
3	Samsung (+Apple)*	2,614	4,664	78%	8.2%	ARM mobile app processors
4	AMD	4,552	3,605	-21%	6.4%	x86 PC, server MPUs
5	Freescale	1,210	1,070	-12%	1.9%	ARM and embedded MPUs
6	Nvidia	591	764	29%	1.4%	ARM mobile app processors
7	TI	510	565	11%	1.0%	ARM mobile app processors
8	ST-Ericsson**	660	540	-18%	1.0%	ARM mobile app processors
9	Broadcom	295	345	17%	0.6%	ARM mobile app processors
10	MediaTek	280	325	16%	0.6%	ARM mobile app processors

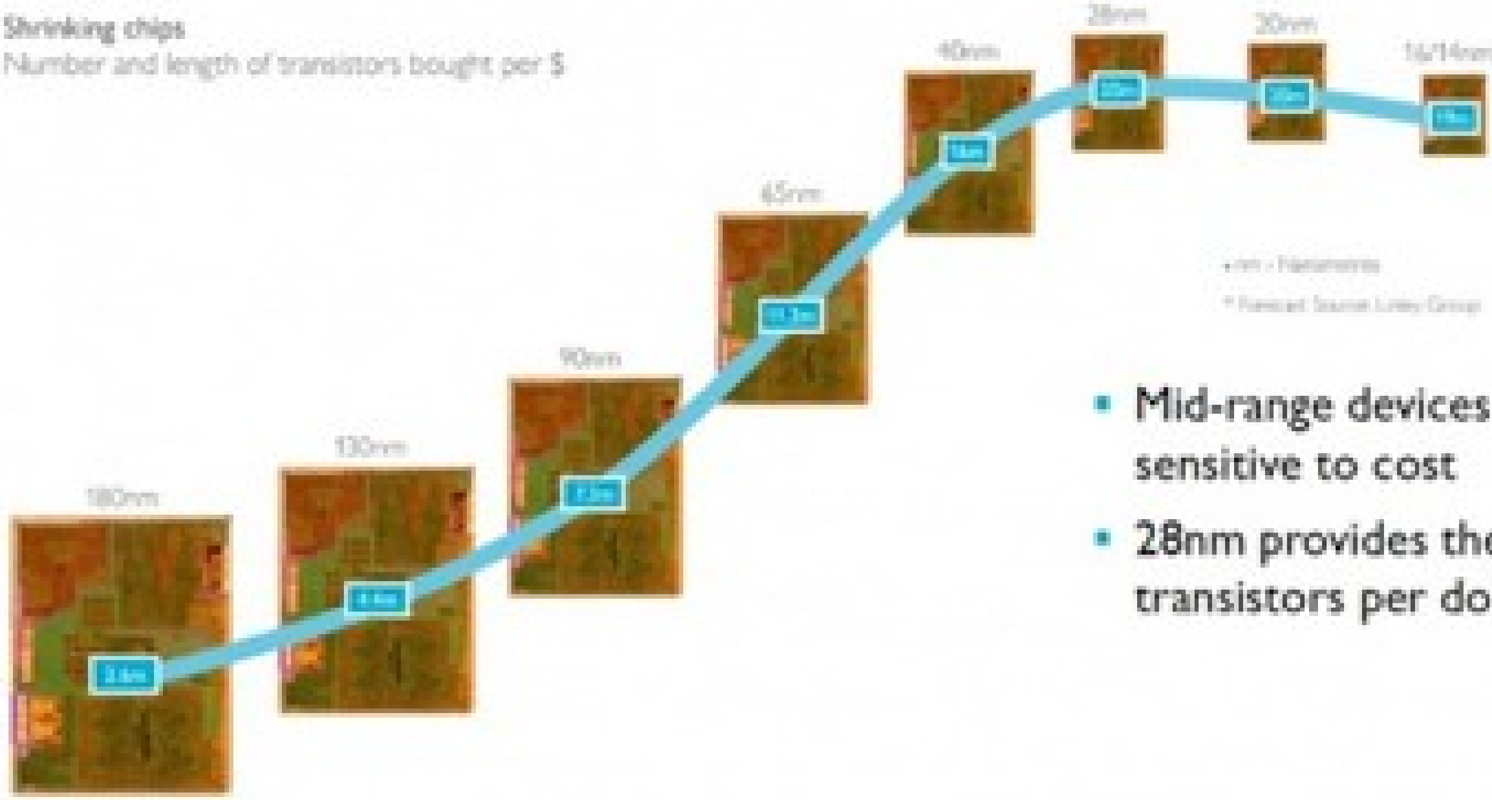
*Includes Apple's custom processors made by Samsung's foundry business.

Source: IC Insights

**Cellphone IC joint venture to be dissovled by STMicroelectronics and Ericsson by 3Q13.

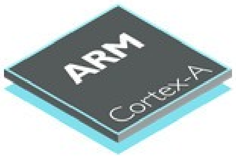
28nm: Optimal Balance of Cost and Power for 2015 Devices

Shrinking chips
Number and length of transistors bought per $



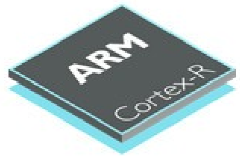
- Mid-range devices are highly sensitive to cost
- 28nm provides the most transistors per dollar

ARM Cortex Series



Cortex-A

Highest performance
Optimized for rich operating systems



Cortex-R

Fast response
Optimized for high-performance, hard real-time applications

Cortex-M

Smallest/lowest power
Optimized for discrete processing and microcontroller

SecurCore

Tamper resistant
Optimized for security applications

Cortex - A

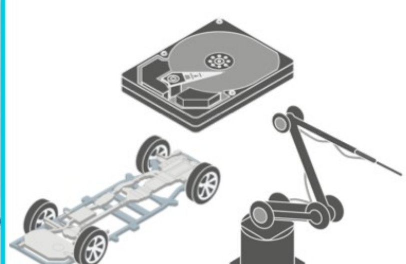
Highest performance

Optimised for rich operating systems

Cortex - R

Fast response

Optimised for high performance, hard real-time applications



Cortex - M

Smallest/lowest power

Optimised for discrete processing and microcontrollers

ARM Cortex-A Series

High performance

Cortex-A73 - 64/32-bit ARMv8-A
2017 Premium Mobile, Consumer

Cortex-A72 - 64/32-bit ARMv8-A
2016 Premium Mobile, Infrastructure & Auto

Cortex-A57 - 64/32-bit ARMv8-A
Proven high-performance

Cortex-A17 - ARMv7-A
High-performance with lower power and smaller area relative to Cortex-A15

Cortex-A15 - ARMv7-A
High-performance with infrastructure feature set

High efficiency

Cortex-A53 - 64/32-bit ARMv8-A
Balanced performance and efficiency

Cortex-A9 - ARMv7-A
Well-established mid-range processor used in many markets

Cortex-A8 - ARMv7-A
First ARMv7-A processor

Ultra-high efficiency

Cortex-A35 - 64/32-bit ARMv8-A
Highest efficiency

Cortex-A32 - 32-bit ARMv8-A
Smallest and lowest power ARMv8-A

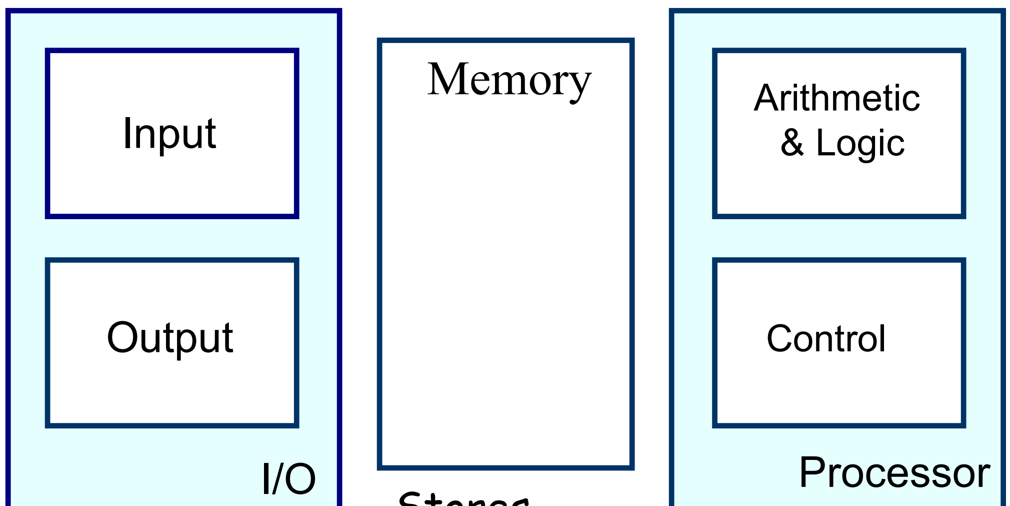
Cortex-A7 - ARMv7-A
Most efficient ARMv7-A CPU, higher performance than Cortex-A5

Cortex-A5 - ARMv7-A
Smallest and lowest power ARMv7-A CPU, optimized for single-core

Functional units of a computer

Input unit accepts information:
- Human operators,
- Electromechanical devices
- Other computers

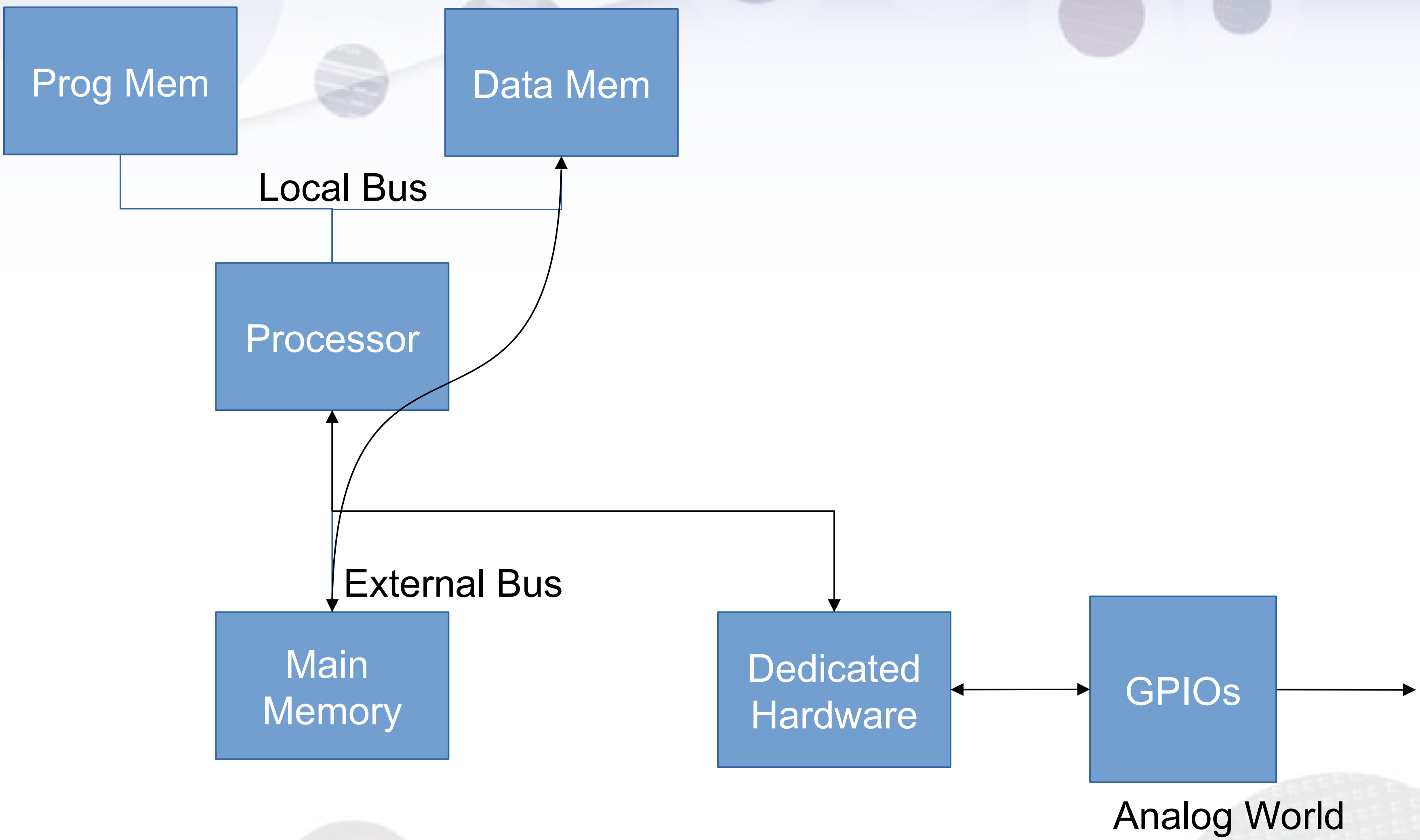
Arithmetic and logic unit(ALU):
- Performs the desired operations on the input information as determined by instructions in the memory



Output unit sends results of processing:
- To a monitor display,
- To a printer

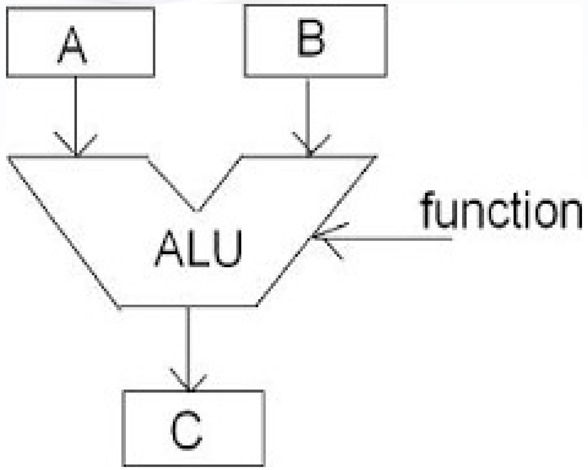
Stores information:
- Instructions,
- Data

Control unit coordinates various actions
- Input,
- Output
- Processing

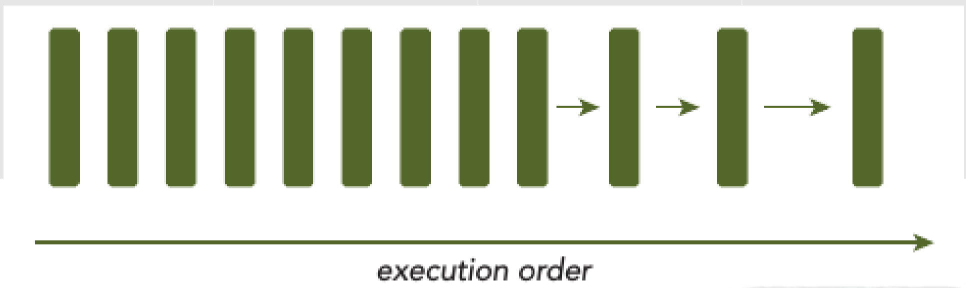


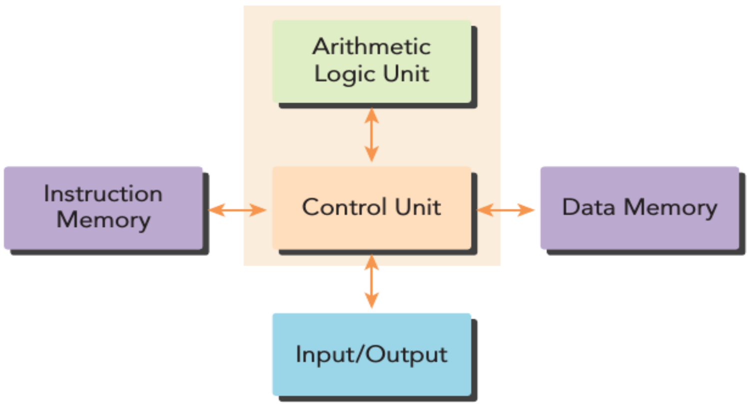
RIPHAH INTERNATIONAL UNIVERSITY

Basic introduction of Microprocessor

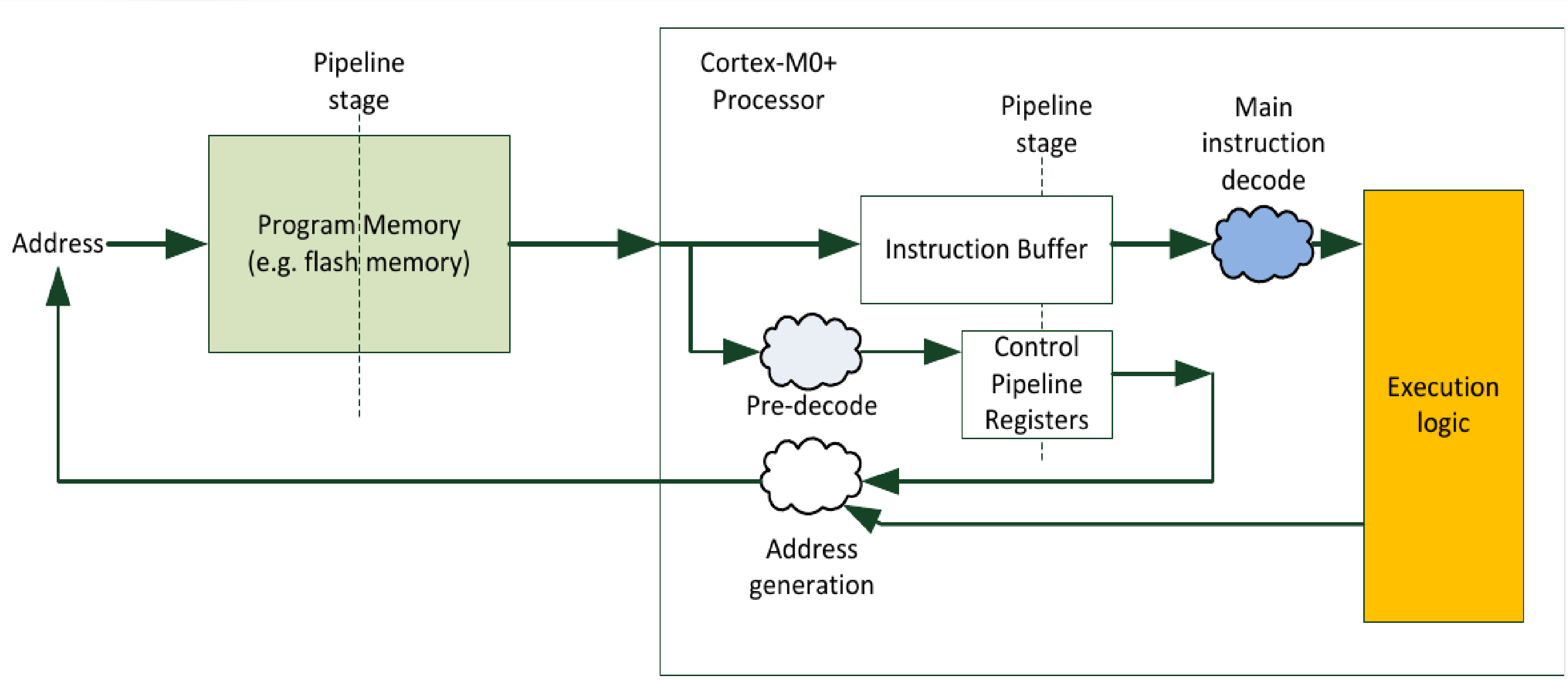


Function	Inputs A	Input B	Output
0	A	B	A+B
1			



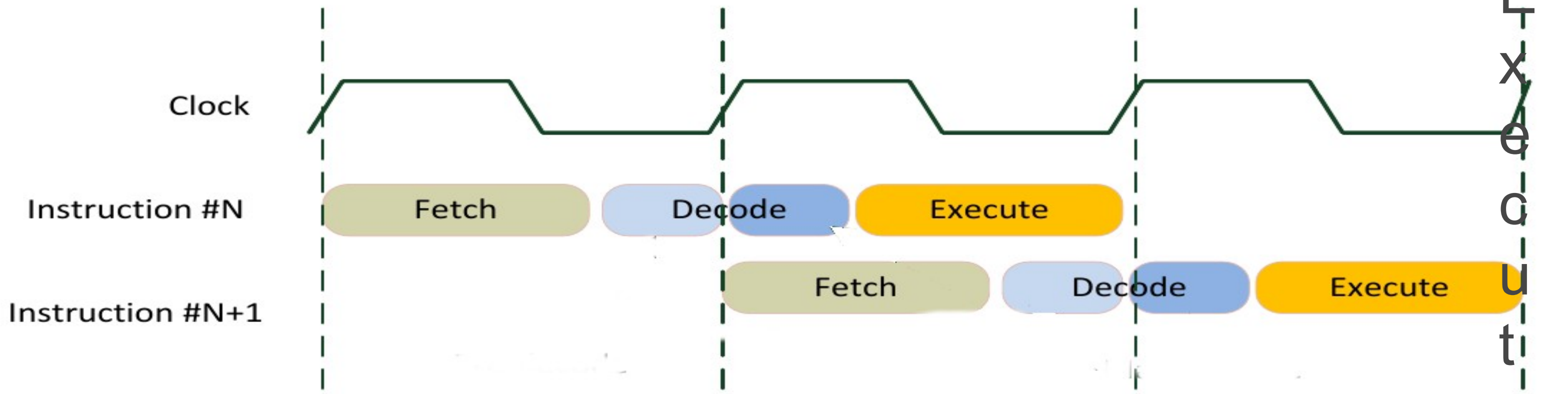


Processor: Performance Improvement



Speed and Performance

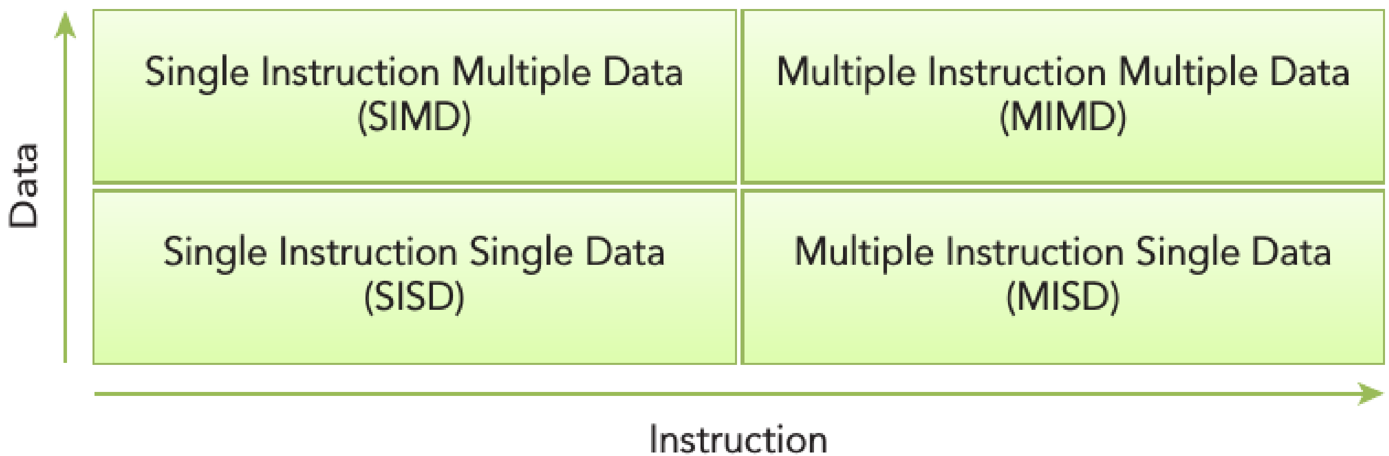
- Fetch
  - Decode
    - Execute
    - Fetch
      - Decode
        - E
x
e
c
u
t
e



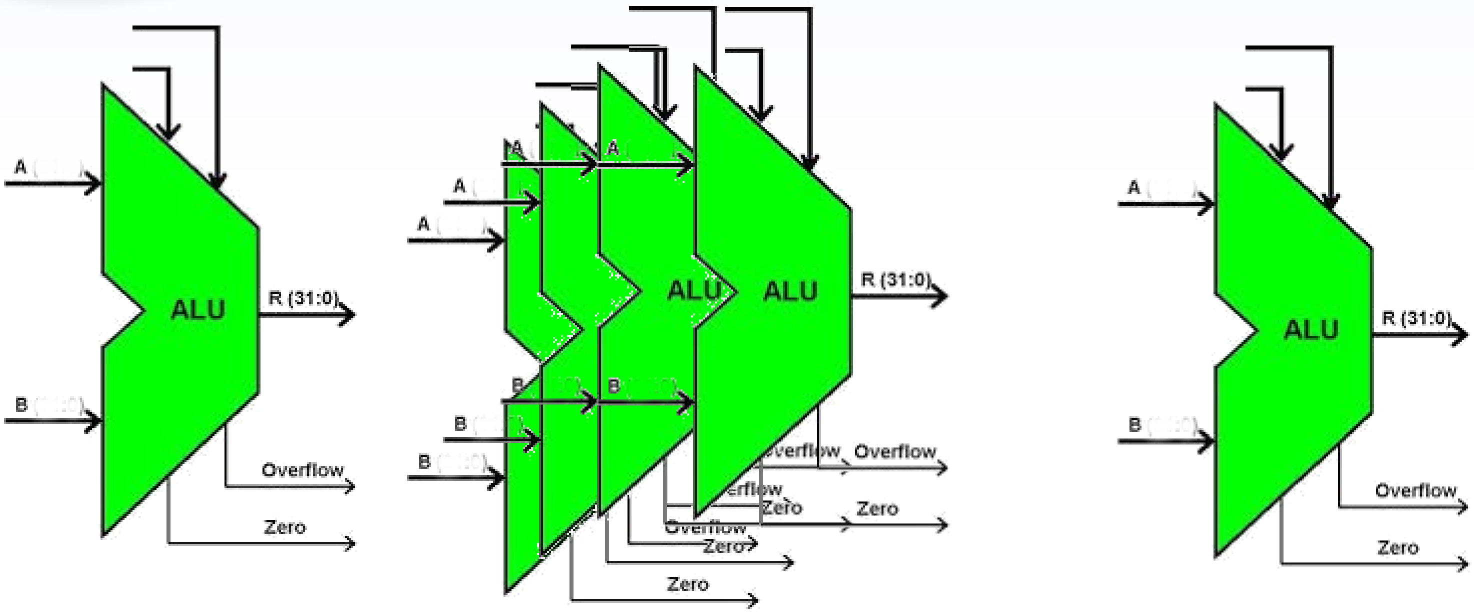
RIPHAH INTERNATIONAL UNIVERSITY

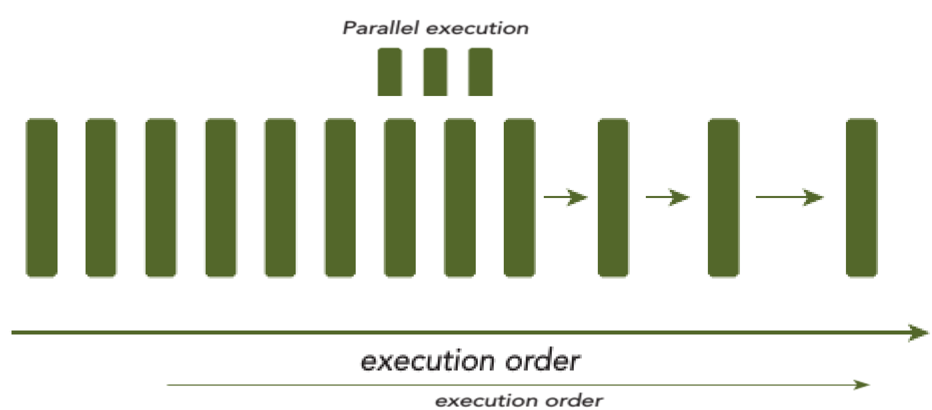
Processor Architectures

- Single Instruction Single Data (SISD)
- Single Instruction Multiple Data (SIMD)
- Multiple Instruction Single Data (MISD)
- Multiple Instruction Multiple Data (MIMD)

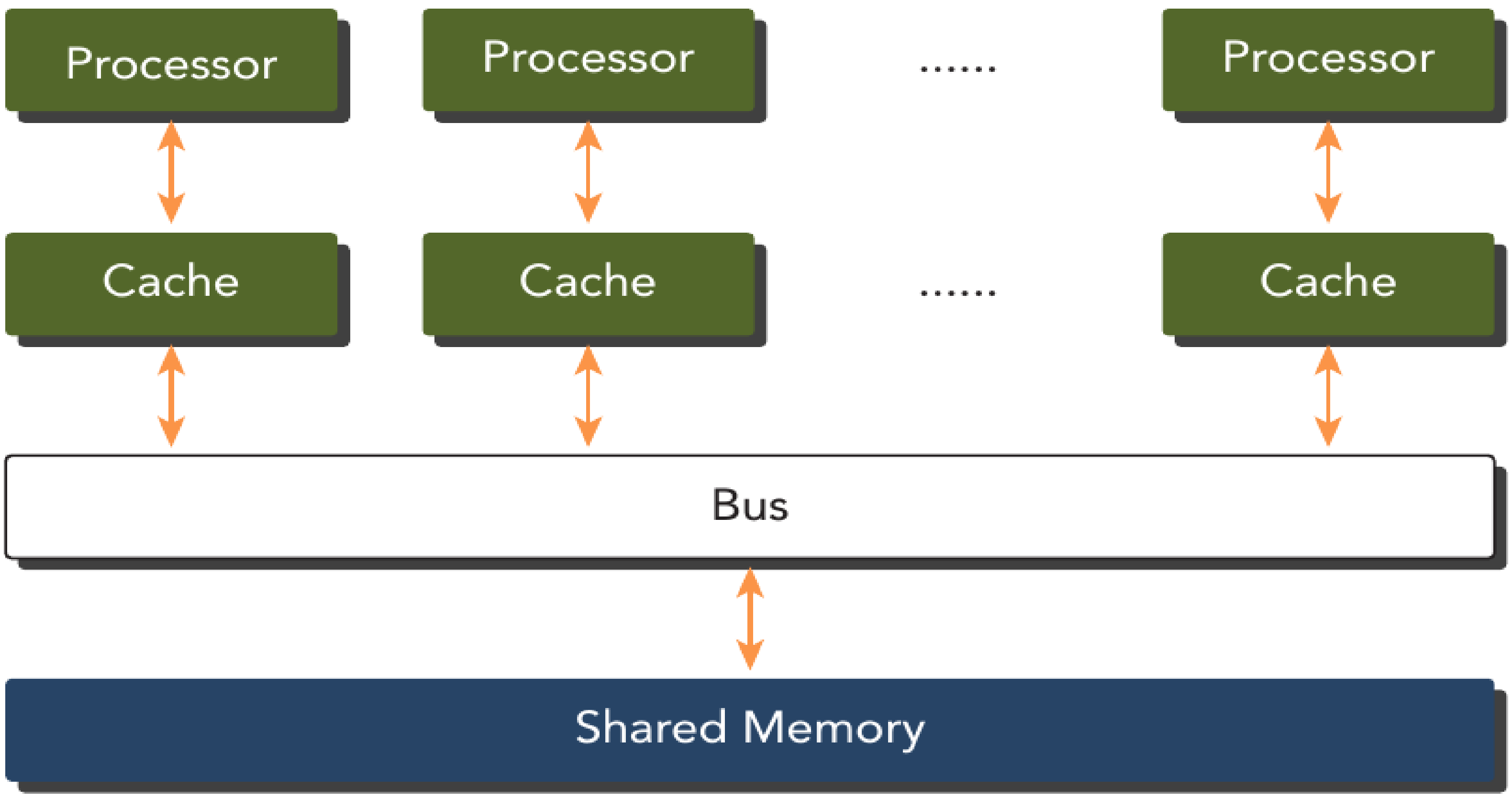


Performance Improvement

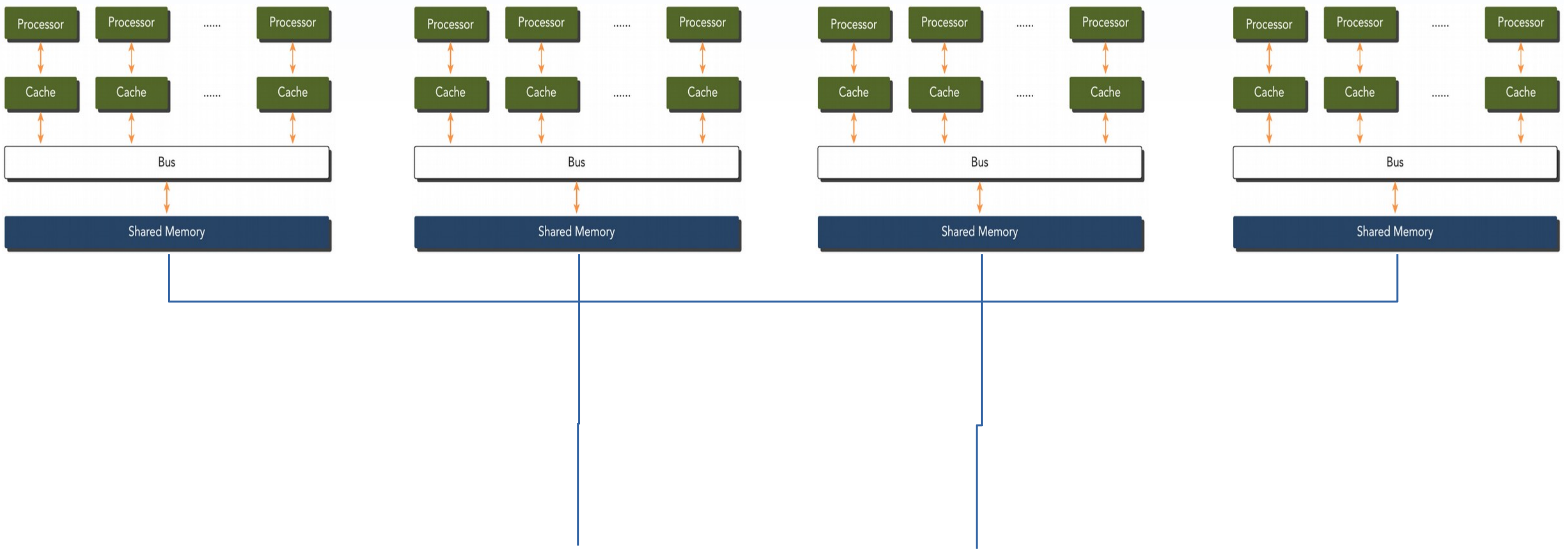




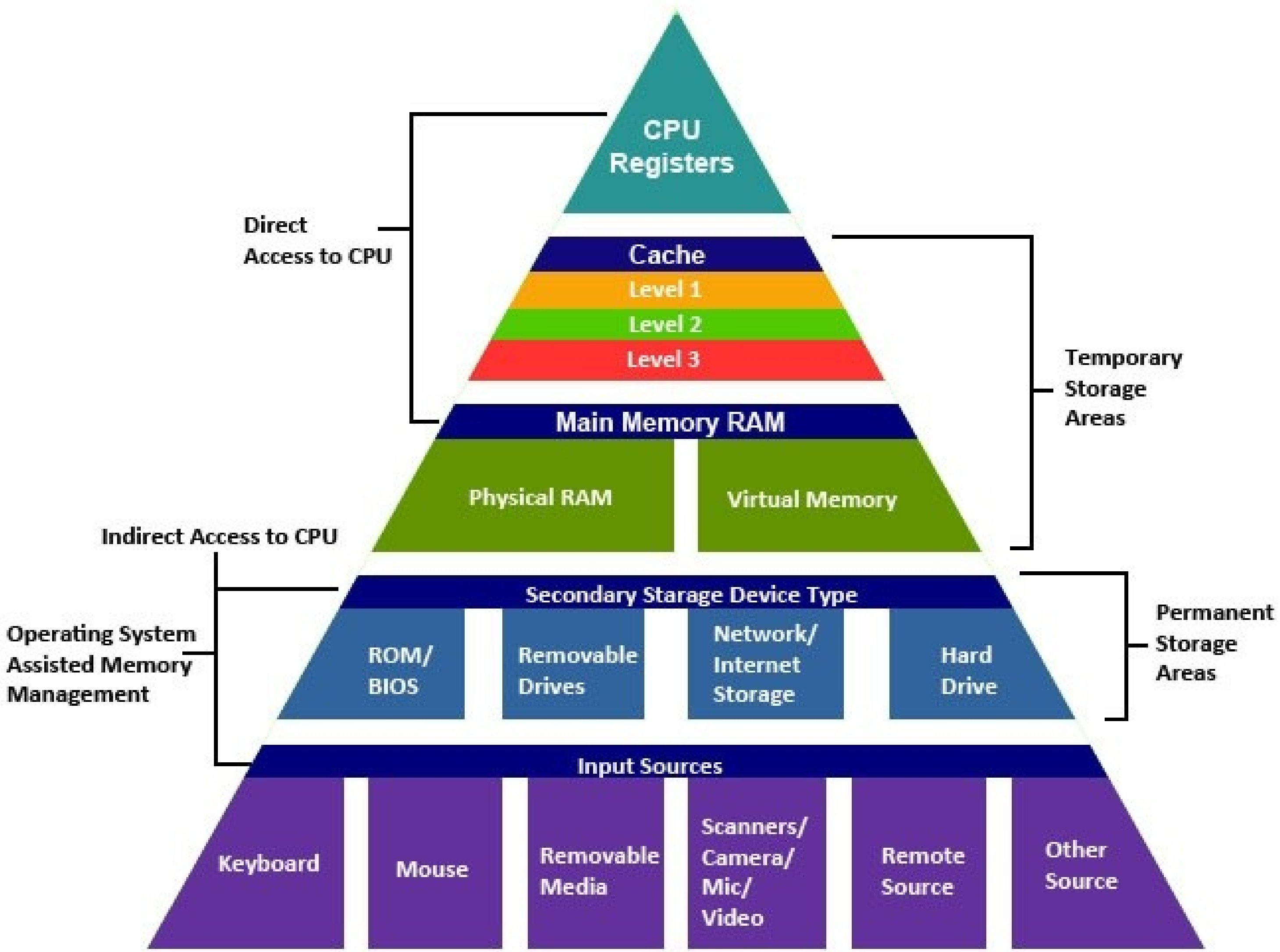
Uni-core, Multi-core CPUs, Clusters, and Grid Computing



Grid Computing



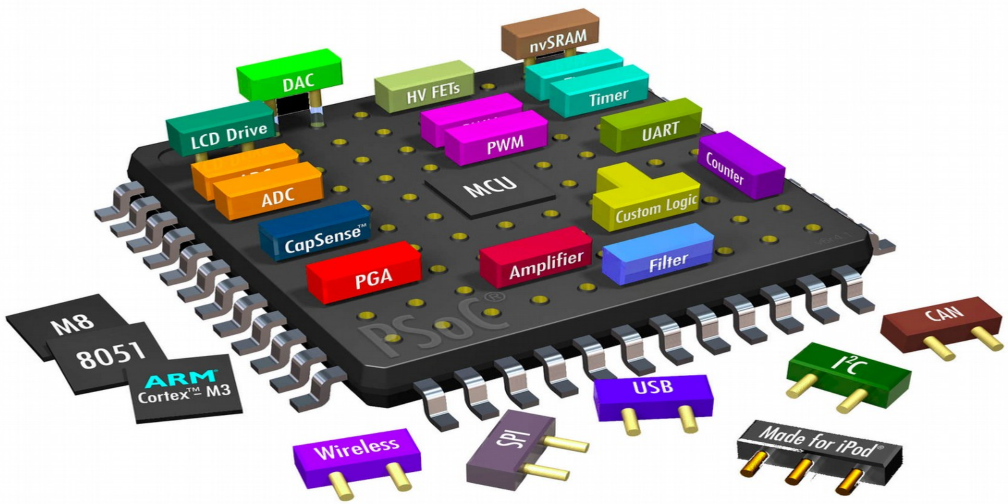
Memories



RIPHAH INTERNATIONAL UNIVERSITY

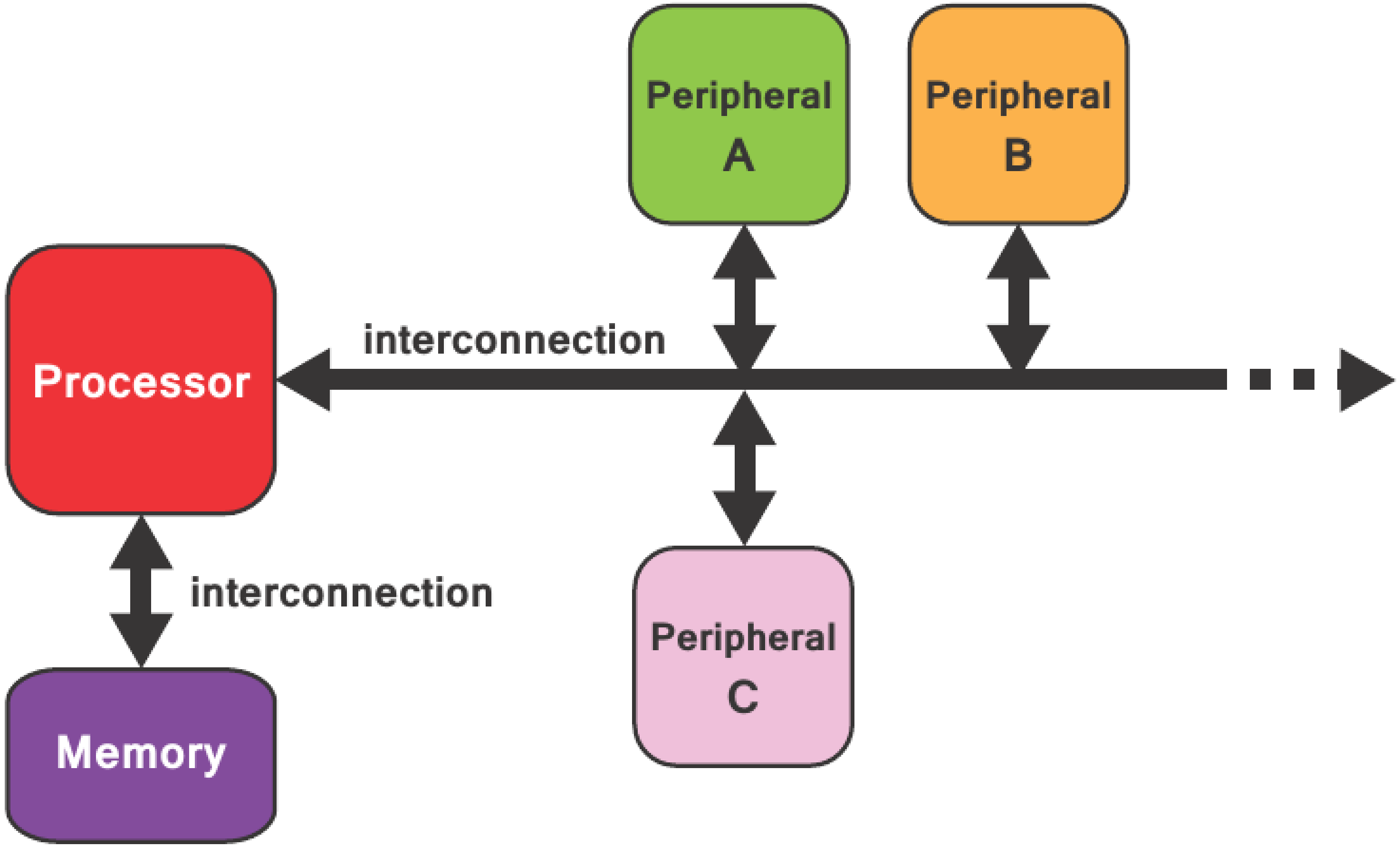
System on Chip Computer Architecture

A system on a chip or system on chip (SoC) is an integrated circuit (IC) that integrates all components of a computer or other electronic system into a single chip. It may contain digital, analog, mixed-signal, and often radio-frequency functions—all on a single chip substrate.

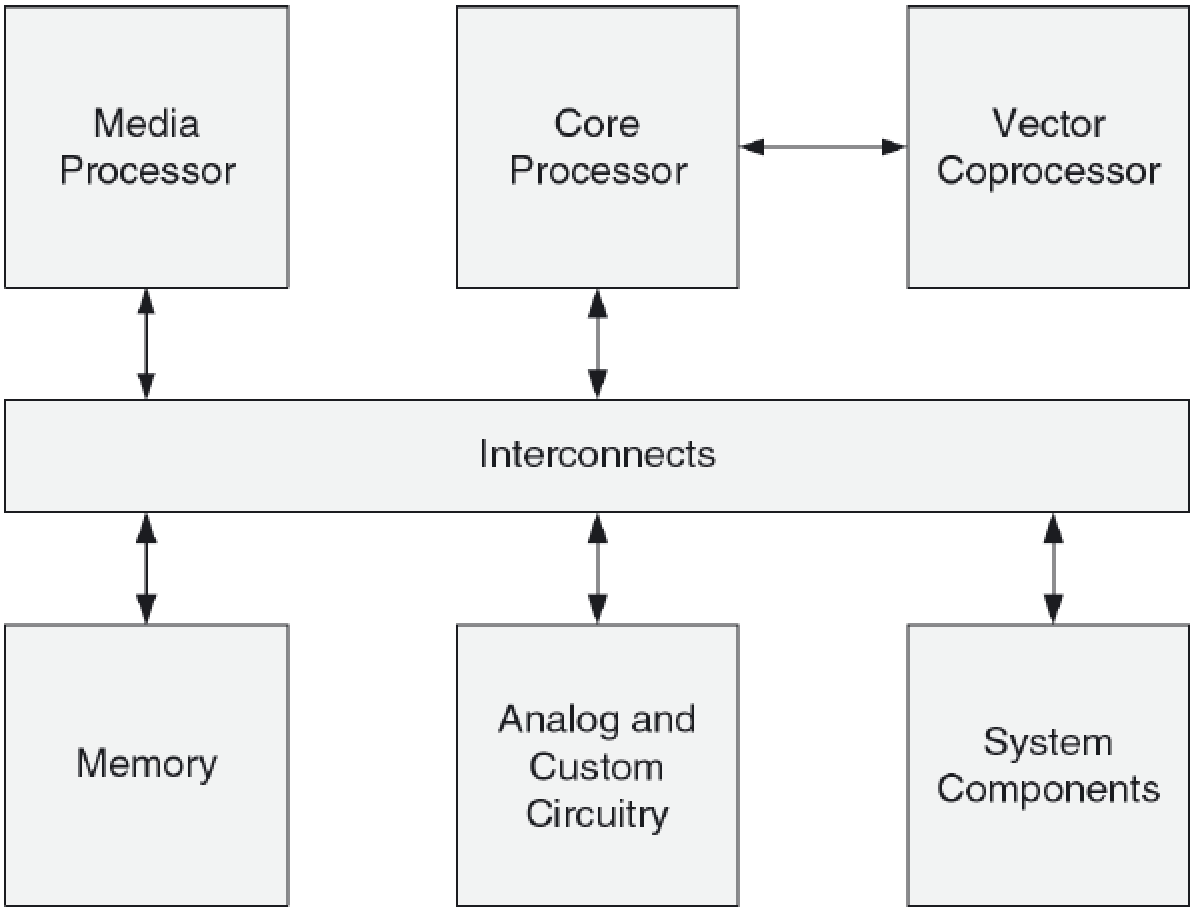


RIPHAH INTERNATIONAL UNIVERSITY

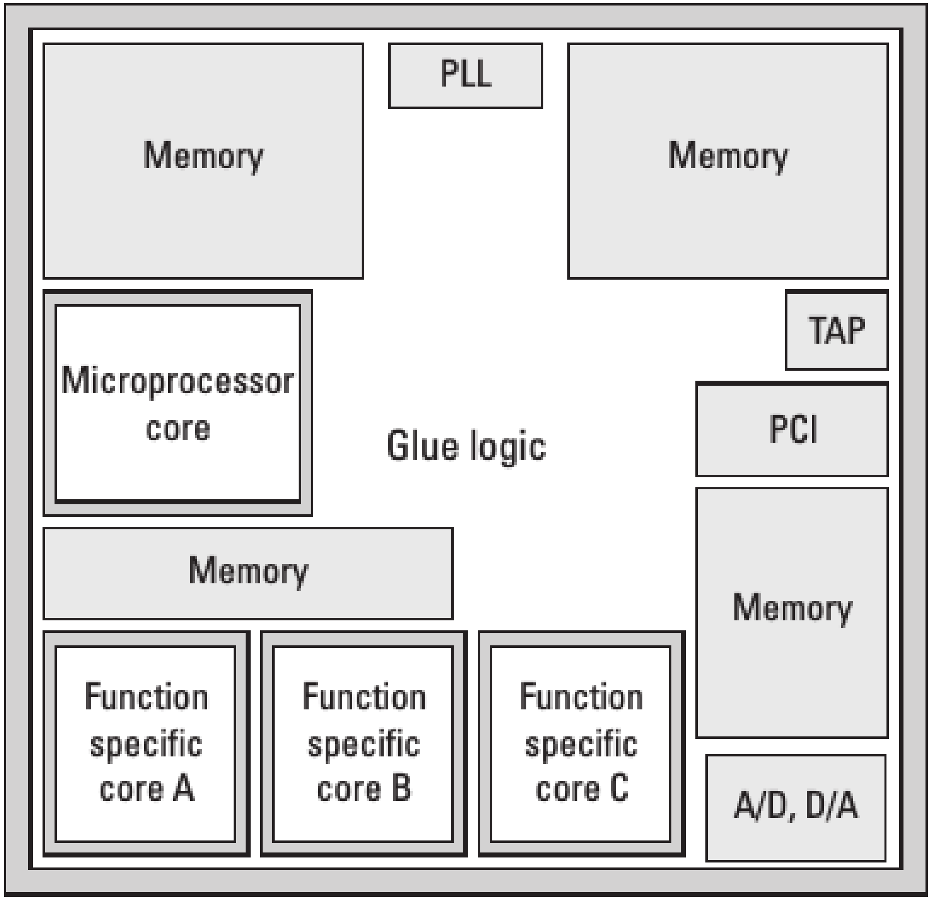
The hardware system architecture of an embedded SoC (simplified)



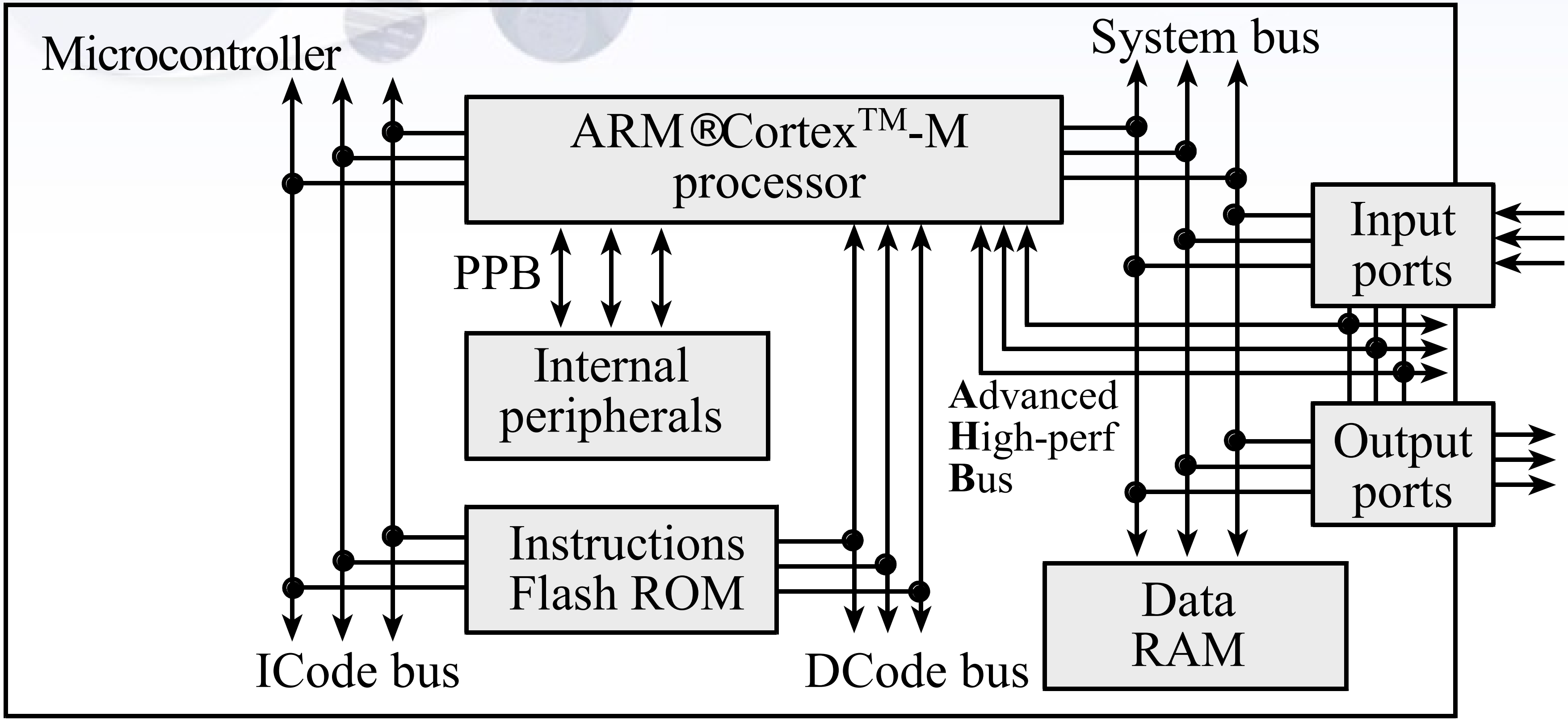
Basic System on Chip Architecture



General architecture of today's embedded core-based system-on-a-chip



ARM Cortex M4-based System



- ARM Cortex-M4 processor
- *Harvard* architecture
  - Different busses for instructions and data

LC3 to ARM - Data Movement

```
LEA  R0, Label                 ;R0 <- PC + Offset to Label
    ADR R0,Label  or  LDR R0,=Label
LD    R1,Label              ; R1 <- M[PC + Offset]
    LDR R0,=Label                  ; Two steps: (i) Get address into R0
    LDRH R1,[R0]                        ; (ii) Get content of address [R0] into R1
LDR  R1,R0,n                ; R1 <- M[R0+n]
    LDRH R1,[R0,#n]
LDI   R1,Label              ; R1 <- M[M[PC + Offset]]
     ; Three steps!!
ST  R1,Label                   ; R1 -> M[PC + Offset]
    LDR R0,=Label                  ; Two steps: (i)Get address into R0
    STRH R1,[R0]                        ; (ii) Put R1 contents into address in R0
STR R1,R0,n                    ; R1 -> M[R0+n]
    STRH R1,[R0,#n]
STI R1,Label                   ; R1 -> M[M[PC + Offset]]
     ; Three steps!!
```

LC3 to ARM – Arithmetic/Logic

```
ADD  R1, R2, R3            ; R1 <- R2 + R3
    ADD R1,R2,R3           ; 32-bit only
ADD  R1,R2,#5              ; R1 <- R2 + 5
    ADD R1,R2,#5           ; 32-bit only, Immediate is 12-bit
AND  R1,R2,R3              ; R1 <- R2 & R3
    AND R1, R2, R3         ; 32-bit only
AND   R1,R2,#1             ; R1 <- Bit 0 of R2
    AND R1, R2, #1         ; 32-bit only
NOT R1,R2                  ; R1 -> ~(R2)
    EOR R1,R2,#-1          ; -1 is 0xFFFFFFFF,
                           ; so bit XOR with 1 gives complement
```

LC3 to ARM – Control

```
BR  Target                    ; PC <- Address of Target
    B Target
BRnzp  Target                 ; PC <- Address of Target
    B Target
BRn  Target                   ; PC <- Address of Target if N=1
    BMI Target                ; Branch on Minus
BRz  Target                   ; PC <- Address of Target if Z=1
    BEQ Target
BRp  Target                   ; PC <- Address of Target if P=1
    No Equivalent
BRnp  Target                  ; PC <- Address of Target if Z=0
    BNE Target
BRzp  Target                  ; PC <- Address of Target if N=0
    BPL Target                ; Branch on positive or zero (Plus)
BRnz  Target                  ; PC <- Address of Target if P=0
    No Equivalent
```

LC3 to ARM – Subs,TRAP,Interrupt

JSR Sub ; PC <- Address of Sub, Return address in R7

BL Sub ; PC<-Address of Sub, Ret. Addr in R14 (Link Reg)

JSRR R4 ; PC <- R4, Return address in R7

BLX R4 ; PC <-R4, Return address in R14 (Link Reg)

RET ; PC <- R7 (Implicit JMP to address in R7)

BX LR ; PC <- R14 (Link Reg)

JMP R2 ; PC <- R2

BX R2 ; PC <- R14 (Link Reg)

TRAP x25 ; PC <- M[x0025], Return address in R7

SVC #0x25 ; Similar in concept but not implementation

RTI ; Pop PC and PSR from Supervisor Stack…

BX LR ; PC <- R14 (Link Reg) [same as RET]

Single Board Computer

A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer.

Types of SBC

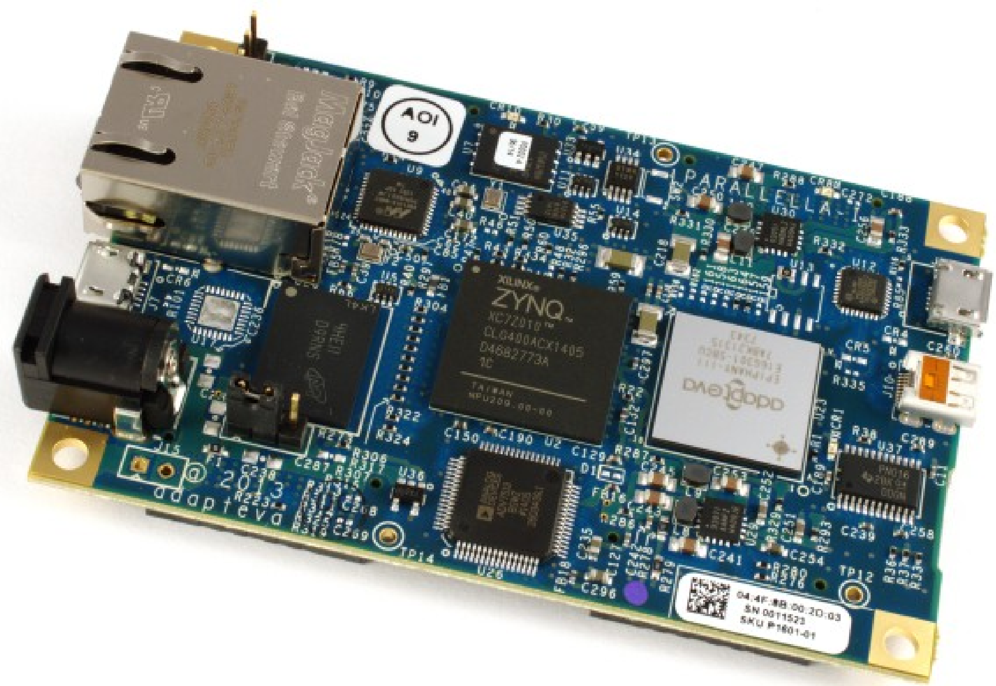
High Performance

Low Power and Low Cost

Parallela

18-core credit card sized computer

- #1 in energy efficiency @ 5W
- 16-core Epiphany RISC SOC
- Zynq SOC (FPGA + ARM A9)
- Gigabit Ethernet
- 1GB SDRAM
- Micro-SD storage
- Up to 48 GPIO pins
- HDMI, USB (optional)
- Open source design files
- Runs Linux



Jetson GPU

GPU: NVIDIA Kepler "GK20a" GPU with 192 SM3.2 CUDA cores (upto 326 GFLOPS)

CPU: NVIDIA "4-Plus-1" 2.32GHz ARM quad-core Cortex-A15 CPU with Cortex-A15 battery-saving shadow-core.

RIPHAH INTERNATIONAL UNIVERSITY

ODROID

* Samsung Exynos5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core CPUs
* Mali-T628 MP6(OpenGL ES 3.1/2.0/1.1 and OpenCL 1.2 Full profile)
* 2Gbyte LPDDR3 RAM PoP stacked
* eMMC5.0 HS400 Flash Storage
* 2 x USB 3.0 Host, 1 x USB 2.0 Host
* Gigabit Ethernet port
* HDMI 1.4a for display
* Size : 83 x 58 x 22 mm approx.(including cooling fa
* Linux Kernel 4.9 LTS

RIPHAH INTERNATIONAL UNIVERSITY

RaspberyPi

Model Name	Release Date	CPU	RAM	Graphics	USB Ports	Power Requirements
Raspberry Pi 1 Model A	February 2012	700 MHz single-core	256 MB	Broadcom VideoCore IV	1	1.5 W
Raspberry Pi 1 Model B	February 2012	700 MHz single-core	256 MB	Broadcom VideoCore IV	2	3.5 W
Raspberry Pi 1 Model A+	February 2013	700 MHz single-core	256 MB	Broadcom VideoCore IV	1	1.0 W
Raspberry Pi 1 Model A+ Revision 2	N/A	700 MHz single-core	512 MB	Broadcom VideoCore IV		1.0 W
Raspberry Pi 1 Model B revision 2	August 2012	700 MHz single-core	512 MB	Broadcom VideoCore IV		3.5 W
Raspberry Pi 1 Model B+	February 2013	700 MHz single-core	512 MB	Broadcom VideoCore IV	4	3.0 W
Raspberry Pi 2 Model B	February 2015	900 MHz quad-core	1 GB	Broadcom VideoCore IV	4	4.0 W
Raspberry Pi Zero	November 2015	1,000 MHz single-core	512 MB	Broadcom VideoCore IV	1 OTG	0.8 W
Raspberry Pi 3 Model B	February 2016	1,200 MHz quad-core	1 GB	Broadcom VideoCore IV with higher clock speeds	4	4.0 W

RaspberryPi Hardware

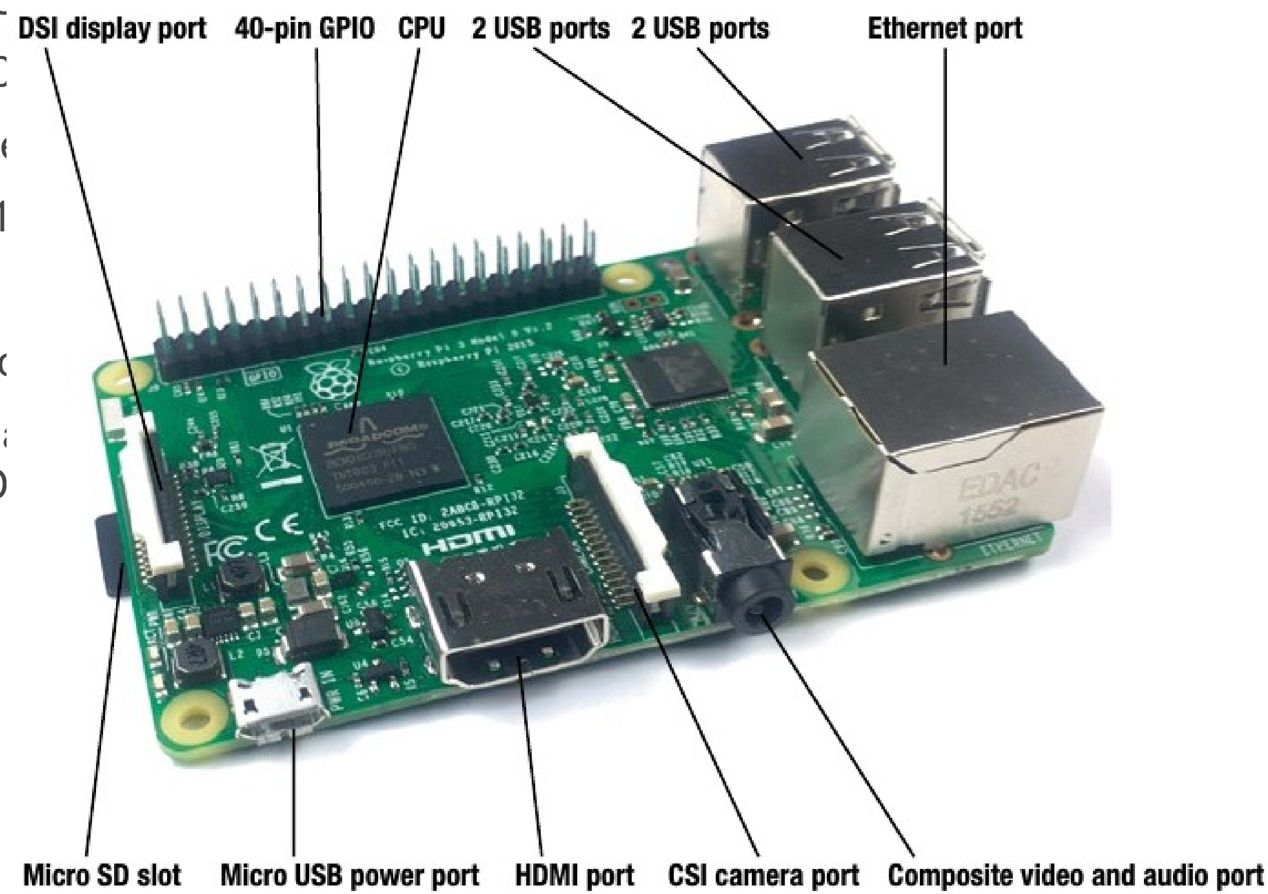
SoC: Broadcom BCM2837

- CPU: 4× ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900 MHz)
- Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- Storage: microSD
- GPIO: 40-pin header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

RaspberryPi Hardware

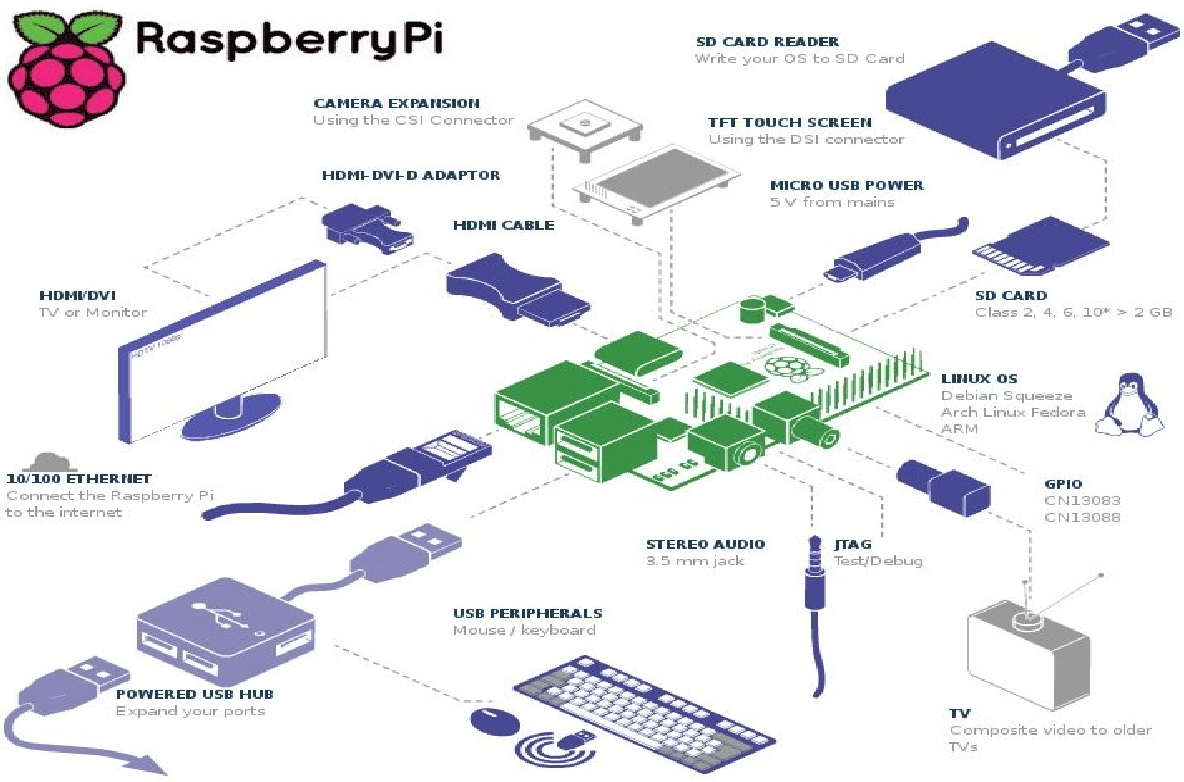
SoC: Broadcom BCM2837

- CPU: 4× ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (9
- Networking: 10/100 Eth
- Bluetooth: Bluetooth 4.1
- Storage: microSD
- GPIO: 40-pin header, p
- Ports: HDMI, 3.5mm an
  Serial Interface (CSI), D



RIPHAH
INTERNATIONAL
UNIVERSITY

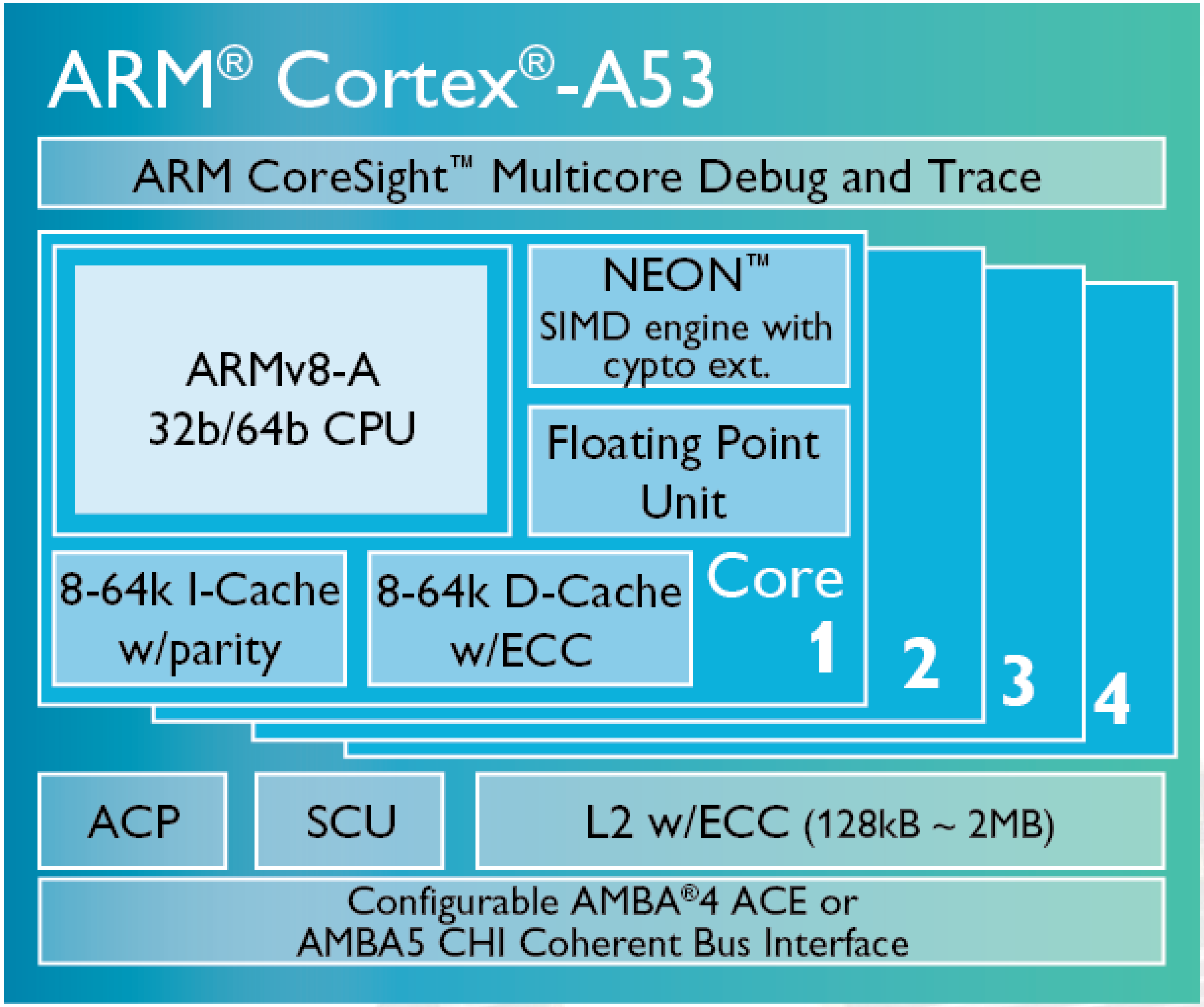
Architecture: RasPi



Using RPi

Because it has an ARM cortex-a processor, it can run the full range of ARM GNU/Linux distributions, including Snappy Ubuntu Core, as well as Microsoft Windows 10.

ARM Cortex-A53 Architecture



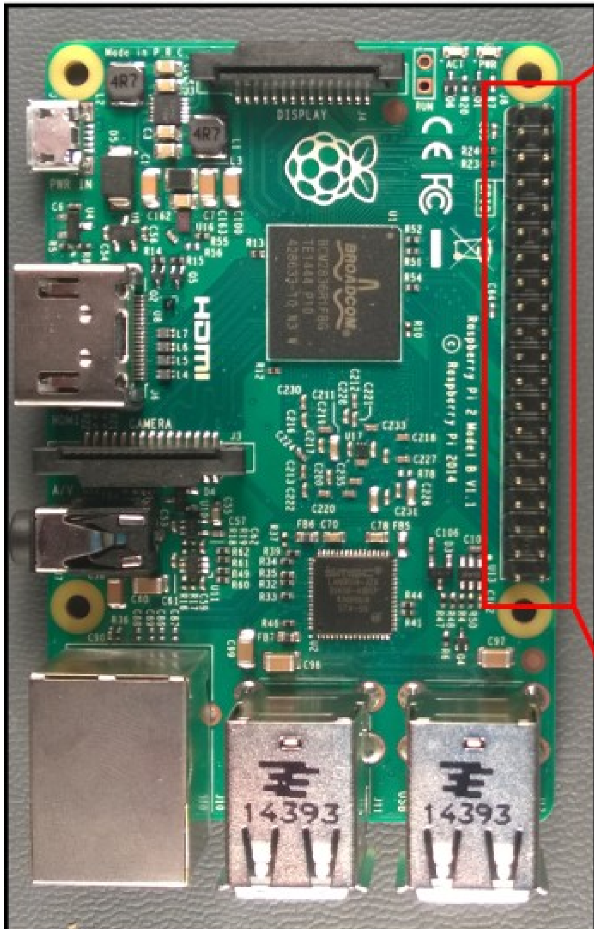
GPIOs: General Purpose Input Outputs

Raspberry Pi2 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

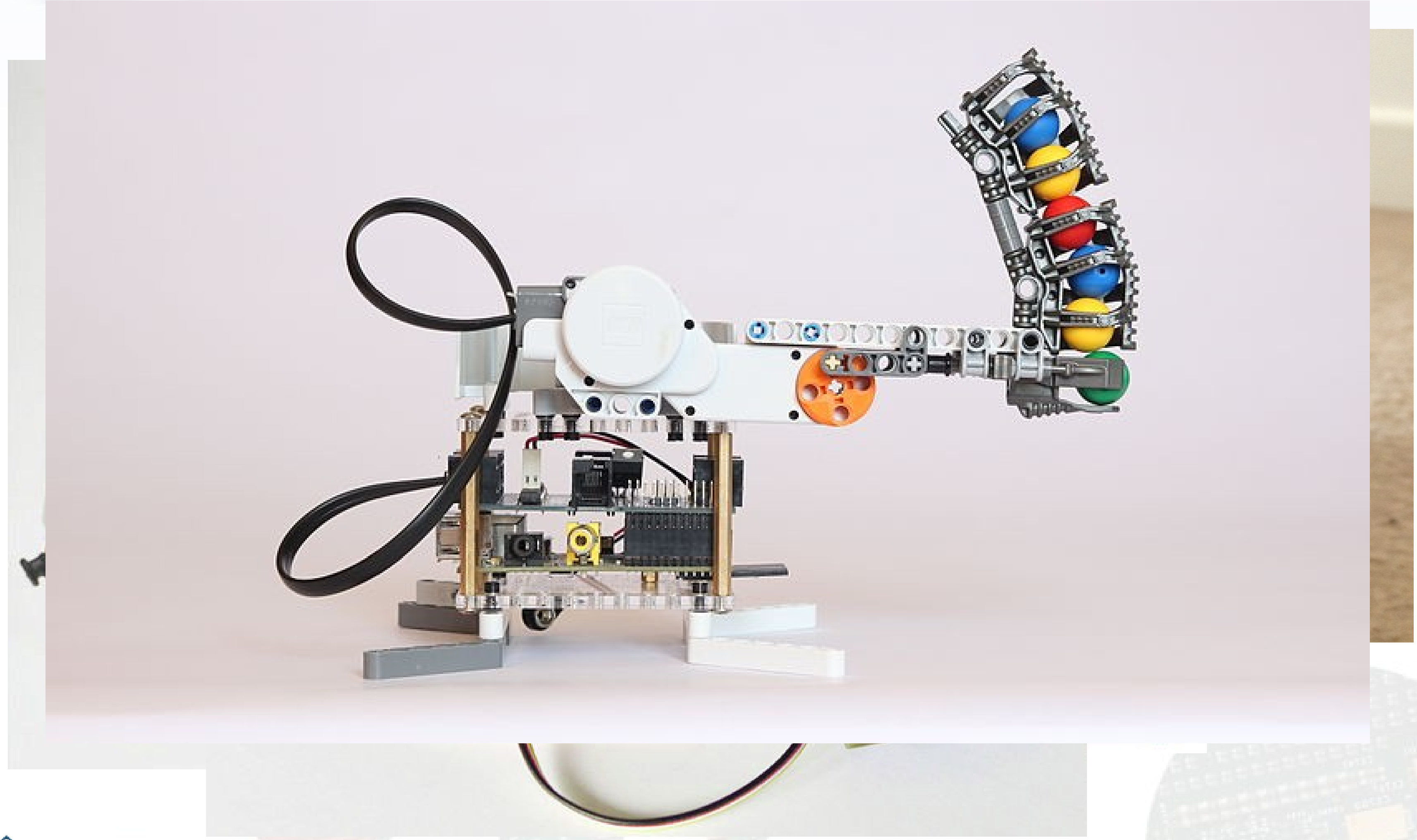
Rev. 1
26/01/2014

http://www.element14.com



Alternate Function					Alternate Function
	3.3V PWR	1	2	5V PWR	
I2C1 SDA	GPIO 2	3	4	5V PWR	
I2C1 SCL	GPIO 3	5	6	GND	
	GPIO 4	7	8	UART0 TX	
	GND	9	10	UART0 RX	
	GPIO 17	11	12	GPIO 18	
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3V PWR	17	18	GPIO 24	
SPI0 MOSI	GPIO 10	19	20	GND	
SPI0 MISO	GPIO 9	21	22	GPIO 25	
SPI0 SCLK	GPIO 11	23	24	GPIO 8	SPI0 CS0
	GND	25	26	GPIO 7	SPI0 CS1
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
SPI1 MISO	GPIO 19	35	36	GPIO 16	SPI1 CS0
	GPIO 26	37	38	GPIO 20	SPI1 MOSI
	GND	39	40	GPIO 21	SPI1 SCLK

Raspi Usage and Applications



RIPHAH INTERNATIONAL UNIVERSITY