 Marwadi University Marwadi Chandarana Group	NAAC A+	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)		Aim: AVR Microcontroller GPIO Programming In C.	
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

Question – 1 :- Perform the comparative analysis of microcontrollers and general-purpose microprocessors along with various criteria to choose an appropriate microcontroller for different applications.

Answer :-

Microcontrollers	General-Purpose Microprocessors
1) Integrated CPU, memory, and I/O peripherals on a single chip	1) CPU with separate memory and I/O interfaces
2) Limited, suitable for specific tasks, lower clock speeds	2) Higher, allows for complex computations, multitasking
3) Optimized for low power consumption	3) Typically higher power consumption
4) Cost-effective due to integration of essential components	4) May incur higher costs, additional components may be required
5) Wide range of built-in peripherals and I/O interfaces	5) Require external components for interfacing
6) Well-suited for real-time applications, deterministic response times	6) May struggle with real-time constraints
7) Limited on-chip memory, suitable for small-scale applications	7) More flexibility in memory expansion, accommodates larger-scale applications

Application requirements :-

- **Processing power:** Consider how much computation is needed.
- **Real-time response:** Determine if quick reaction to inputs is crucial.
- **Power consumption:** Decide if the device needs to be battery-powered.
- **Cost constraints:** Define your budget for the microcontroller (MCU).
- **Peripheral integrations:** Assess if specific features like timers, ADC, or communication modules are required.



Development environment:

- **Ease of use:** Consider if you are comfortable with complex development tools.
- Software availability: Check if libraries and tools are available for your chosen MCU.
- Future scalability:

- **Growth potential:** Consider if your application might need more features in the future.

Examples:

- **Simple sensor project:** Choose a low-cost MCU with an integrated ADC and communication interface.

 Marwadi University Marwadi Chandarana Group		Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

Question – 2 :- Perform the comparative analysis of microcontrollers and general-purpose microprocessors along with various criteria to choose an appropriate microcontroller for different applications.


Answer :-

A Brief History of the AVR Microcontroller

- The AVR microcontroller has its origins in the collaborative efforts of Alf-Egil Bogen and Vegard Wollan, two students at the Norwegian Institute of Technology (NTH). Their basic architecture design caught the attention of Atmel, leading to its acquisition and further development by the company in 1996.
- The acronym "AVR" has various interpretations. While Atmel officially maintains it as simply a product name, some speculate it could stand for "Advanced Virtual RISC" or even represent the initials of its original designers, Alf and Vegard RISC.
- AVR microcontrollers come in various types, with the notable exception of AVR32, which is a 32-bit microcontroller. The majority are 8-bit microprocessors, capable of processing data in 8-bit chunks. This means larger data must be fragmented into 8-bit pieces for CPU processing.
- One challenge with AVR microcontrollers is their lack of 100% software compatibility when transitioning between different families. Moving from one family to another may necessitate recompilation of programs and potential adjustments to register locations. For instance, programs written for the ATtiny25 may require recompilation and register modifications to run on an ATmega64.
- AVRs are typically categorized into four main groups: Mega, Tiny, Special Purpose, and Classic. This text focuses on the Mega family due to its widespread use and versatility. Specifically, attention is given to the ATmega32 variant, known for its power, availability, and compatibility with DIP packages, making it particularly suitable for educational purposes. Once proficient with the Mega family, understanding and adapting to other families becomes relatively straightforward.

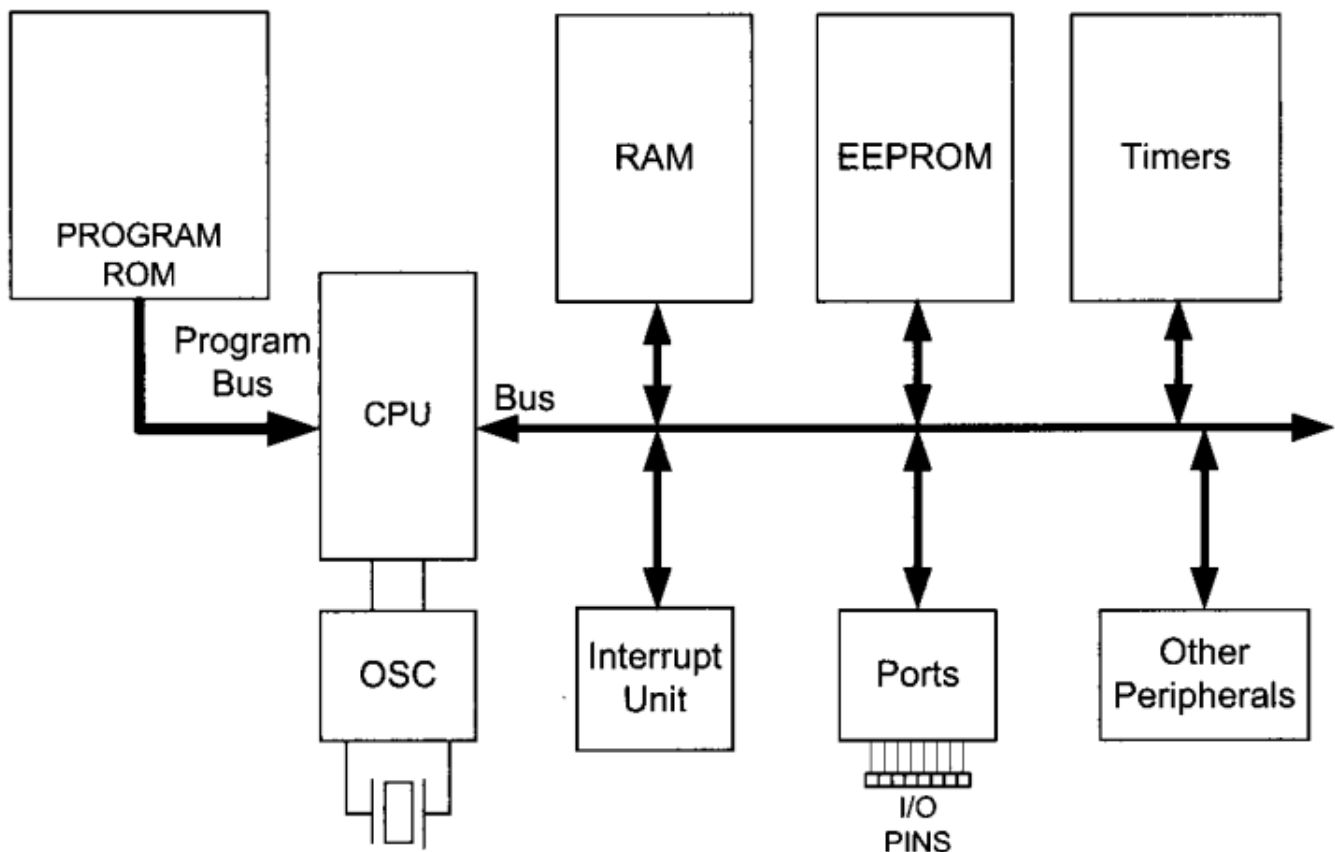
AVR features

- The AVR is an 8-bit RISC single-chip microcontroller with Harvard architecture that comes with some standard features such as on-chip program (code) ROM, data RAM, data EEPROM, timers and I/O ports. See Figure 2. Most AVRs have some additional features like ADC, PWM, and different kinds of serial interface such as USART, SPI, I2C (TWI), CAN, USB, and so on.

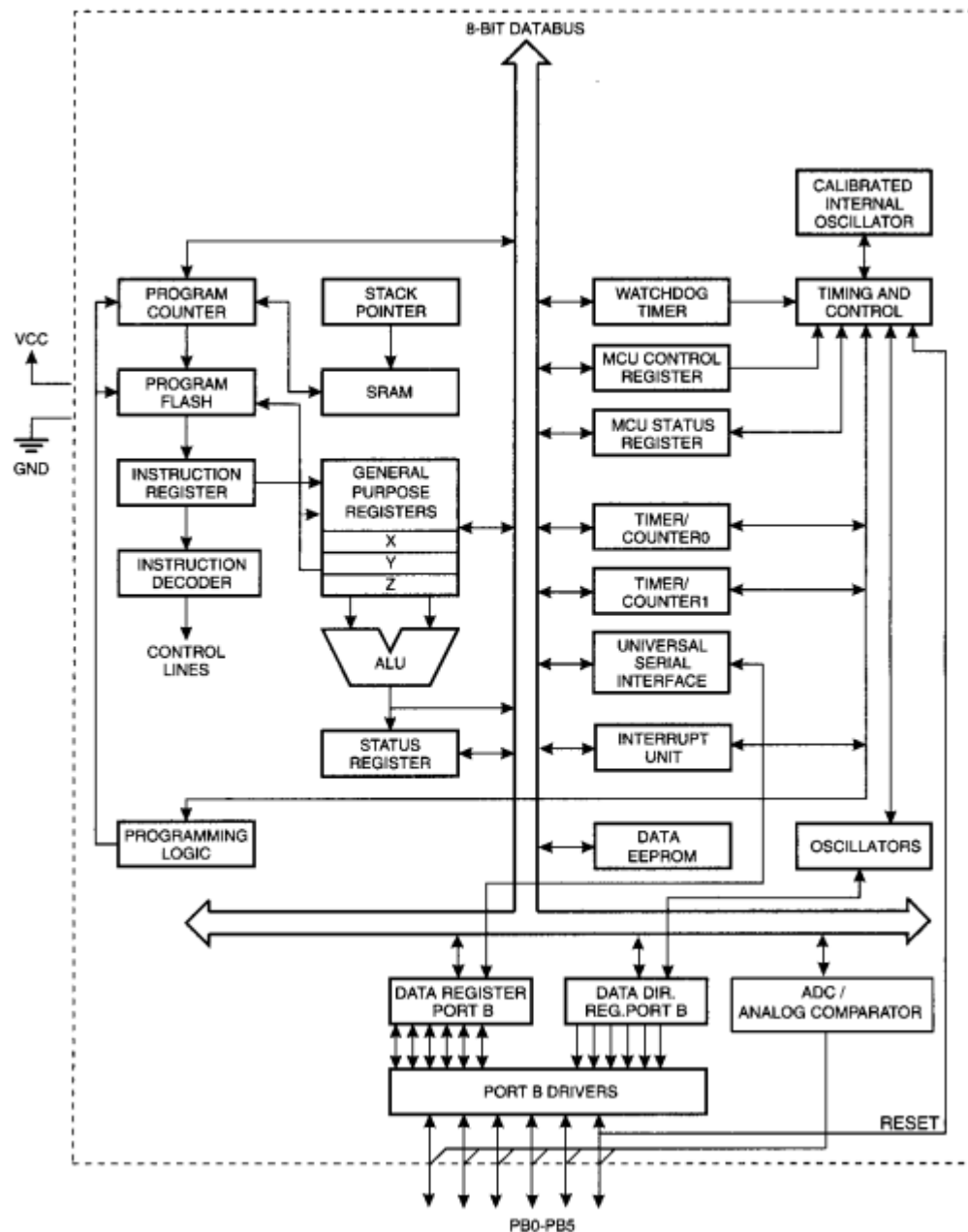
 Marwadi University Marwadi Chandarana Group	NAAC A+	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	


AVR Micro Controller Programm ROM :-

- In microcontrollers, the ROM is used to store programs and for that reason, it is called program or code ROM. Although the AVR has 8M (megabytes) of program (code) ROM space, not all family members come with that much ROM installed. The program ROM size can vary from 1K to 256K at the time of this writing, depending on the family member. The AVR was one of the first microcontrollers to use on-chip Flash memory for program storage. The Flash memory is ideal for fast development because Flash memory can be erased in seconds compared to the 20 minutes or more needed for the UV-EPROM. A discussion of the various types of ROM is given in Chapter 0, if you need to refresh your memory on these important memory technologies.



 Marwadi University Marwadi Chandarana Group	NAAC 	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

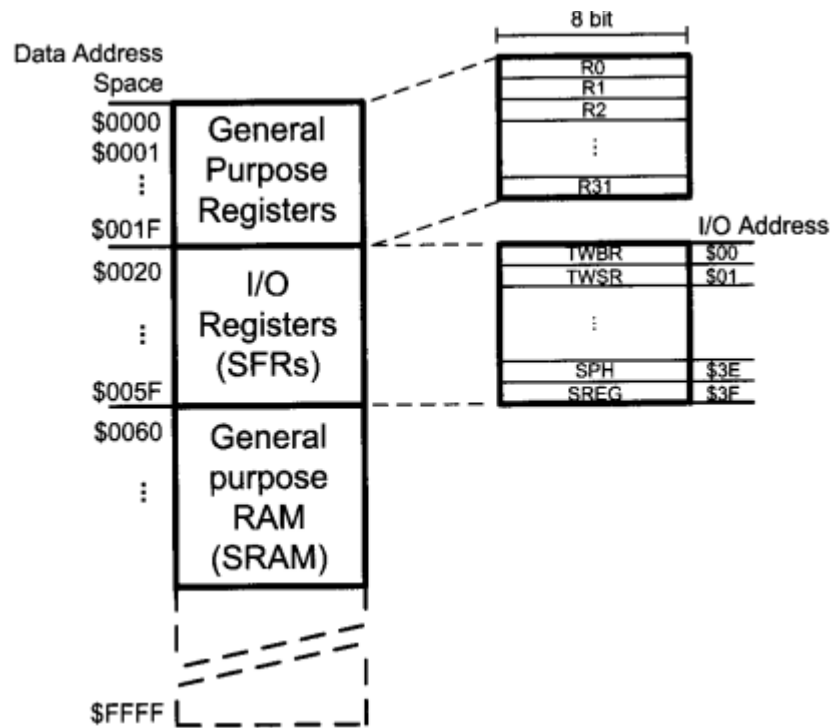



 Marwadi University Marwadi Chandarana Group	NAAC A+	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assigment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

Question – 3 :- Show following things with suitable diagrams in context to data memory of AVR

- 1) The Data memory for AVR with no extended I/O Memory

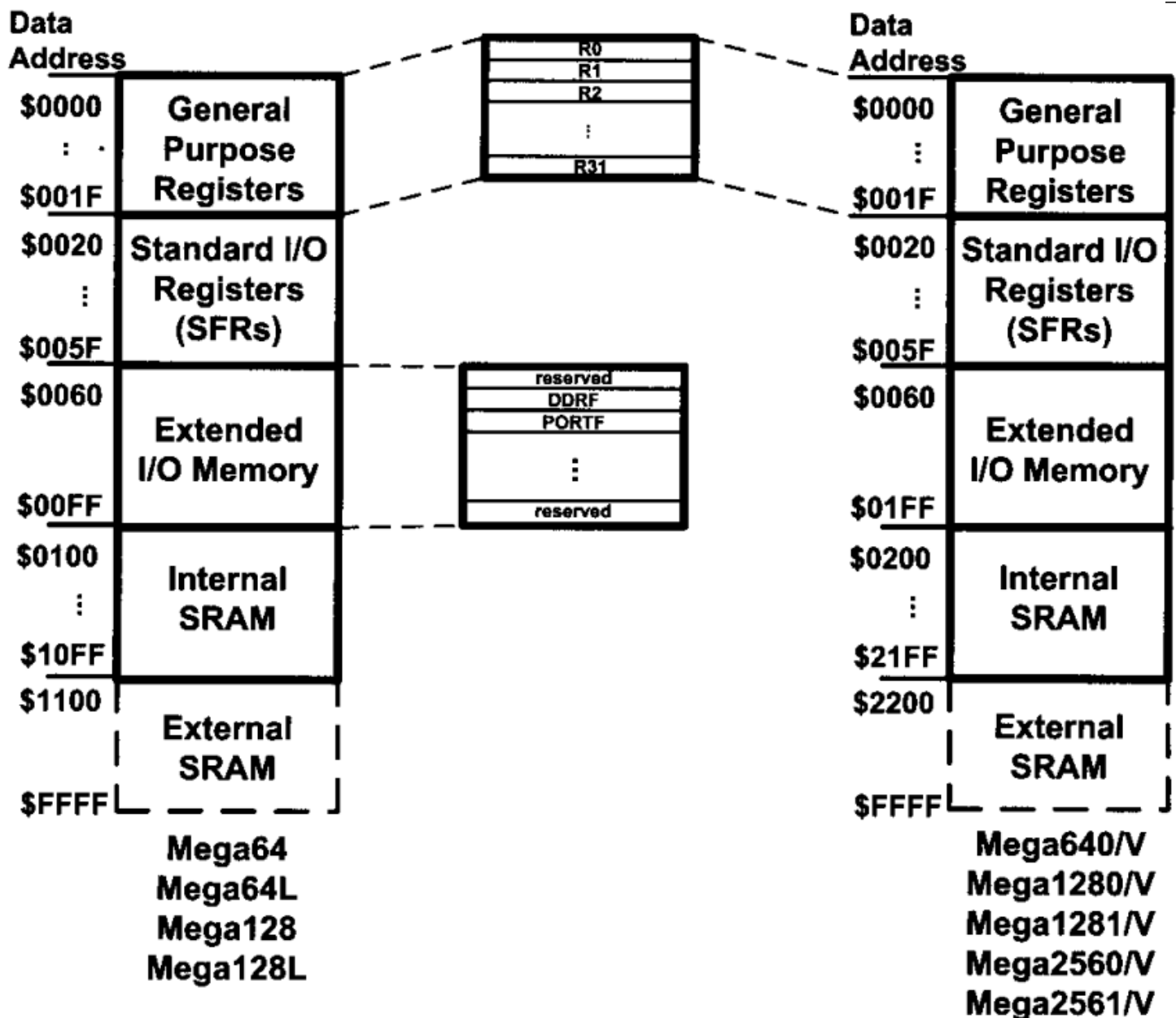
Answer :-





 Marwadi University Marwadi Chandarana Group	NAAC A+	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

2) The data memory for AVR with extended I/O memory

Answer :-



 Marwadi University Marwadi Chandarana Group		Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)		Aim: AVR Microcontroller GPIO Programming In C.	
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

3) Analyze data memory of different variants of AVR family

AVR Chip	Data Memory (Bytes)	I/O Registers (Bytes)	SRAM (Bytes)	General Purpose Registers
ATtiny25	224	64	128	32
ATtiny85	608	64	512	32
ATmega8	1120	64	1024	32
ATmega16	1120	64	1024	32
ATmega32	2144	64	2048	32
ATmega128	4352	64+160	4096	32
ATmega2560	8704	64+416	8192	32

Question – 4 :- Give the significance of following instructions in terms of their size, formats execution, operation, etc:

Answer :-

1. **LDI (Load Immediate):**


- **Size:** LDI is a two-byte instruction.
- **Format:** LDI loads an immediate 8-bit constant into a register, using the next byte as the immediate value. The instruction opcode is followed by the register number and the immediate value.
- **Execution:** LDI executes quickly as it directly loads data into a register without memory access.
- **Operation:** Useful for initializing variables, setting flags, and performing arithmetic operations with constants efficiently.

2. **LDS (Load Direct from Data Space):**

- **Size:** LDS is a three-byte instruction.
- **Format:** LDS loads data from a specific data memory address into a register. It requires two bytes to specify the 16-bit data memory address.
- **Execution:** LDS requires memory access, so it might be slower compared to LDI, but still efficient for accessing non-immediate data.
- **Operation:** Valuable for accessing variables and data structures stored in data memory, enabling manipulation of external data.

3. **STS (Store Direct to Data Space):**

- **Size:** STS is also a three-byte instruction.
- **Format:** STS stores the contents of a register into a specific data memory address. It requires two bytes to specify the 16-bit data memory address.
- **Execution:** Similar to LDS, STS requires memory access and may incur slight overhead.

 Marwadi University Marwadi Chandarana Group	NAAC A+	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	



-
- **Operation:** Essential for storing computed results, variables, and data structures back into data memory, facilitating data persistence.

4. IN (Input from I/O Port):

- **Size:** IN is a one-byte instruction.
- **Format:** IN reads data from a specified I/O port into a register. The instruction opcode is followed by the register number and the I/O port address.
- **Execution:** IN executes efficiently by directly reading from hardware I/O ports without involving memory access.
- **Operation:** Critical for interfacing with peripherals such as sensors, displays, and communication modules, enabling data acquisition from external sources.


5. OUT (Output to I/O Port):

- **Size:** OUT is also a one-byte instruction.
- **Format:** OUT writes data from a register to a specified I/O port. The instruction opcode is followed by the register number and the I/O port address.
- **Execution:** Similar to IN, OUT executes efficiently by directly manipulating hardware I/O ports.
- **Operation:** Crucial for controlling external hardware components, including LEDs, motors, and communication interfaces, facilitating data output to external devices.

 Marwadi University Marwadi Chandarana Group		Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

Question – 5 :- Perform the comparative analysis of RISC and CISC.

RISC(Reduced Instruction Set Computer)	CISC(Complex Instruction Set Computer)
1) Simplified instruction set with a focus on basic operations	1) Extensive instruction set supporting complex operations and addressing modes
2) Instructions typically executed in one clock cycle	2) Instructions may require multiple clock cycles for execution
3) RISC processors often have deeper pipelines, enabling higher clock speeds	3) CISC processors may have shorter pipelines, potentially limiting clock speeds
4) Regular memory access patterns, often with separate load and store instructions	4) Variable memory access patterns, with instructions capable of accessing memory directly
5) RISC instructions tend to be more concise, leading to smaller code size	5) CISC instructions can be more complex, resulting in larger code size
6) Emphasis on simpler hardware design, with less complex instruction decoding logic	6) More complex hardware design, including microcode and sophisticated instruction decoding
7) Typically achieves high performance through pipelining and optimization for common operations	7) Performance can vary depending on the specific implementation and workload
8) Generally lower power consumption due to simpler instruction execution	8) May consume more power due to complex instruction execution and decoding

 Marwadi University Marwadi Chandarana Group	NAAC A+	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Microcontroller and Interfacing (01CT0403)	Aim: AVR Microcontroller GPIO Programming In C.		
Assignment :- 1	Date:- 09-02-2024	Enrollment No:- 92200133030	

Question – 6 :- Perform the comparative analysis of Harvard and Von-neuman architectures.

Harvard Architecture	Von Neumann Architecture
1) Separate memory units for instructions and data	1) Single memory unit for both instructions and data
2) Simultaneous access to instruction and data memory	2) Sequential access to memory for both instructions and data
3) Independent pathways for instruction and data fetch	3) Common pathway for instruction and data fetch
4) Potential for higher performance due to simultaneous access	4) Typically lower performance due to sequential access
5) Instructions can be fetched and executed simultaneously	5) Instructions fetched and executed sequentially
6) Requires additional hardware for separate instruction and data memory	6) Simplified hardware design with unified memory
7) May provide better support for pipelining and parallel processing	7) Less flexible for pipelining and parallelism
8) Can complicate programming due to separate memory spaces	8) Simplified programming model with unified memory
9) Many modern microcontrollers and DSPs use Harvard architecture	9) Most desktop and laptop computers use Von Neumann architecture