

MIPS Implementations in Computer Organizations

This presentation explores the MIPS architecture, its instruction set, execution cycle, and its implementation in modern computer systems. We'll delve into pipelining techniques, hazards, and the advancements in MIPS microarchitecture.

Introduction to MIPS Architecture

Reduced Instruction Set Computing (RISC)

MIPS is a RISC architecture that emphasizes simplicity and efficiency, focusing on a small set of instructions with fixed-length formats. It has been widely used in embedded systems, networking devices, and even supercomputers.

Load-Store Architecture

MIPS is a load-store architecture, where data can only be accessed through load and store instructions. This simplifies the instruction set and allows for faster execution.

Instruction Set

Format	Arithmetic	R-type	Memory	Control	J-type
Entlode - 0et8	32	25980	090	25882	Bat5
Entlode - Jet4	10	25778	090	29933	Bat7
Gatlode - Jef9	12	29441	084	25387	Rat3
Entlode - Jef9	30	29831	064	20322	Bat3
Gatlode - Def5	10	28972	064	39922	Bat7
EnMode - 0et9	10	26573	085	26922	Rat6
Entlode - 0et5	78	28872	0831	39973	Bat7
Gatlode - 0et6	70	25003	0981	25377	Bat7
Entlode - Jef7	75	25878	0991	29708	Bat7
Entlode - 0et8	33	26802	0962	95272	Bat6
Entlode - 0ef2	55	35599	0955	16837	Bat5
EnMode - Jet8	38	28903	9982	19935	Bat5
EnMode - 0et7	88	35723	0991	15572	Bat6
EnMode - 0et7	16	15833	1991	15547	Bat7
Entlode - 0et1	12	35974	1991	15203	Bat7
Entlode - 0et3	75	16604	1986	16644	Bat3
Entlode - Jet2	70	19937	3992	37103	Bat7
Entlode - 0et5	30	18632	3092	35142	Bat7
Entlode - 0et3	54	10731	9996	33209	Rat7
Entlode - 0et4	36	19931	9996	36532	Bat7
Gatlode - 0et9	106	335438	9954	667.73	Bat2
EnMode - 0et7	307	352.158	9992	664037	Rat4
Entlode - 5et6	335	155488	9972	364481	Bat7
EnMode - 3et6	163	157738	9965	564398	Bat7
EnMode - 3et1	152	152794	9978	352138	Bat3
Gatlode - Jet5	155	120337	5027	355348	Bat7
EnMode - Jef7	184	353338	9996	353338	Bat5
CaMode - Jet5	157	140831	9904	332431	Bat7
CaMode - 0et7	156	357338	8061	355328	Bat7
EnMode - 3et5	367	39409	9991	35893	Bat3
Gatlode - 0et7	311	79551	5001	79931	7993

MIPS Instruction Set and Addressing Modes

Instruction Types

MIPS instructions are categorized into arithmetic, logical, data transfer, control flow, and system instructions. These categories simplify the design and implementation of the processor.

Addressing Modes

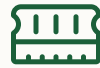
MIPS supports various addressing modes, including register direct, immediate, register indirect, and base-offset addressing. These modes offer flexibility in accessing data in memory.

MIPS Register File and Memory Organization



Register File

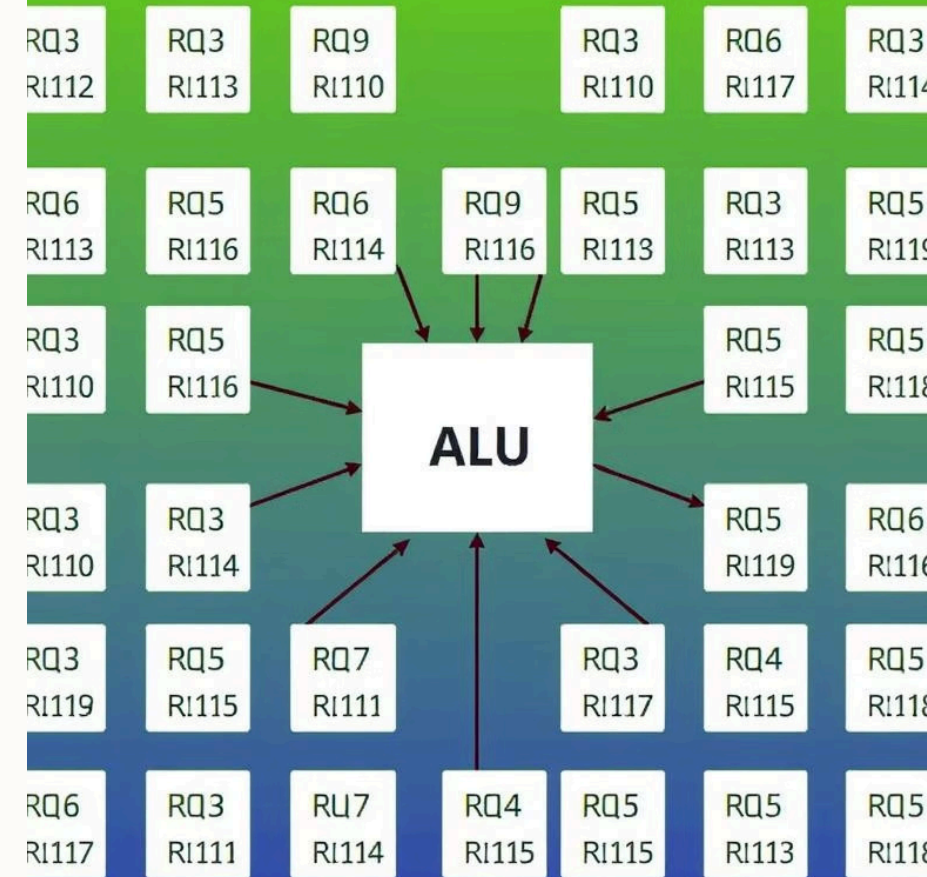
MIPS has 32 general-purpose registers, providing fast access to frequently used data. These registers are used to store temporary values and program variables.

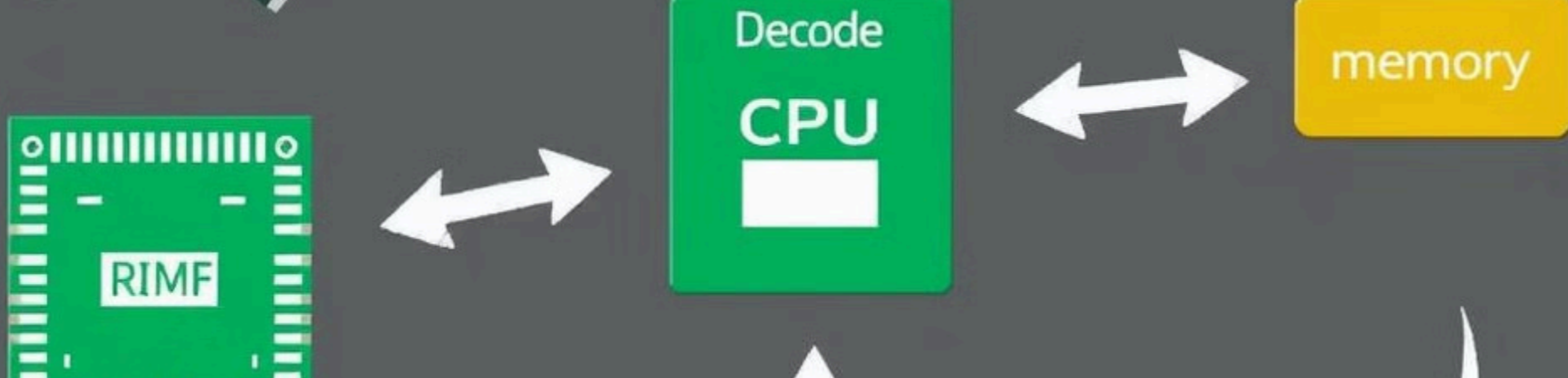


Memory Organization

MIPS memory is organized into a linear address space, with each byte having a unique address. This structure facilitates efficient memory management and data access.

MIPS Register file





MIPS Instruction Execution Cycle

1

Fetch: The instruction is fetched from memory and placed into the instruction register.

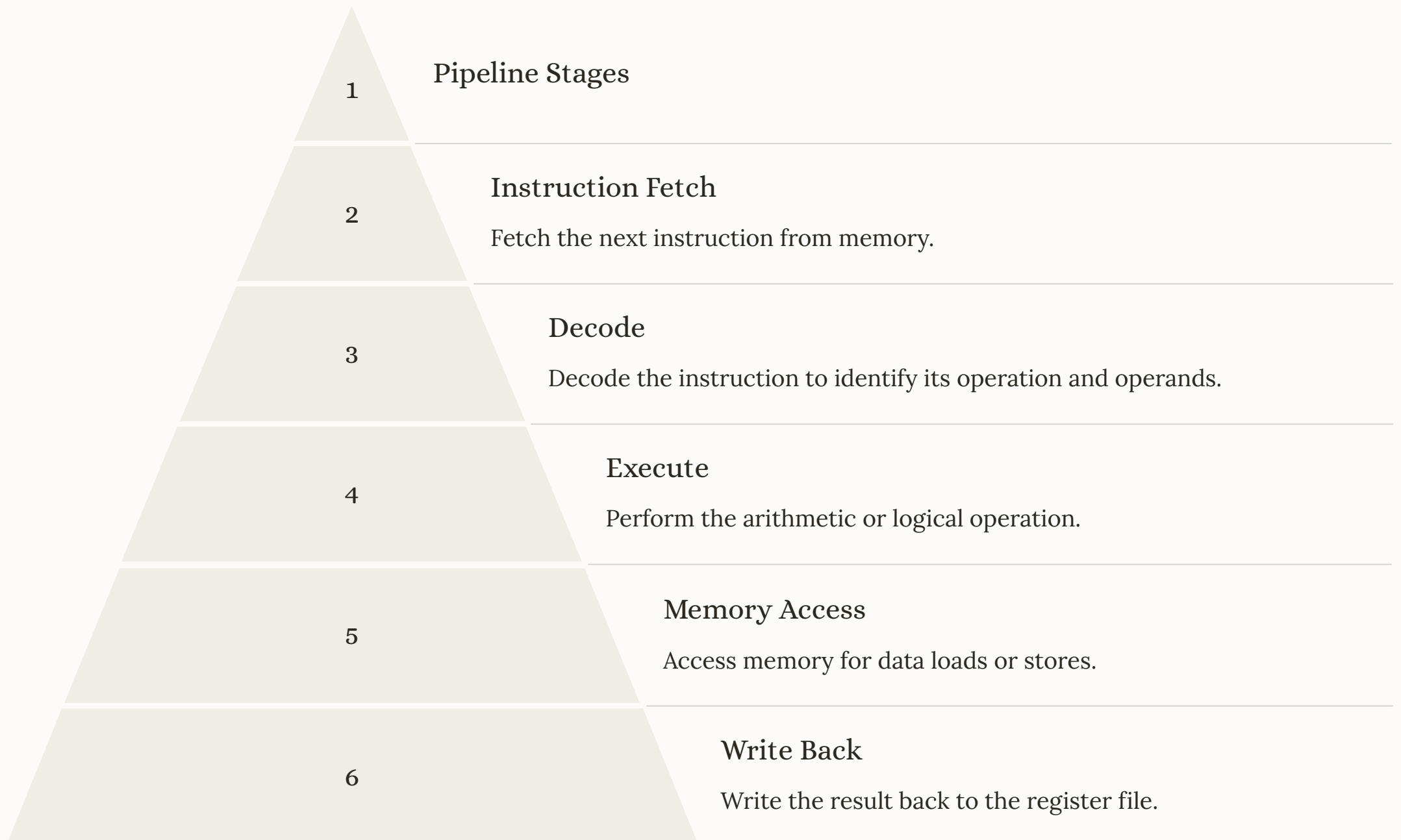
2

Decode: The instruction is decoded, identifying the operation and operands.

3

Execute: The operation is performed, with the result being stored in the register file or memory.

Pipelining in MIPS Processors



Hazards and Stalls in MIPS Pipelines

1

Data Hazards

Occur when an instruction depends on the result of a previous instruction that hasn't completed yet.

2

Control Hazards

Occur when a branch instruction changes the flow of control, disrupting the pipeline's sequential execution.

3

Stalls

Pipeline stalls introduce delays to resolve hazards, ensuring the correct sequence of instructions.

MIPS Pipeline by data hazard)

acceeti.e:ttlisting

decceti.e:ttlisting

acceeti.e:ttlisting

acceeti.e: t(1510)

deceeti.c:ttlisting

dcidetie.e:tl(linl

dcceeti.e:ttlisting

dcceeti.e:tl(lin)

deceeti.e:ftlisting

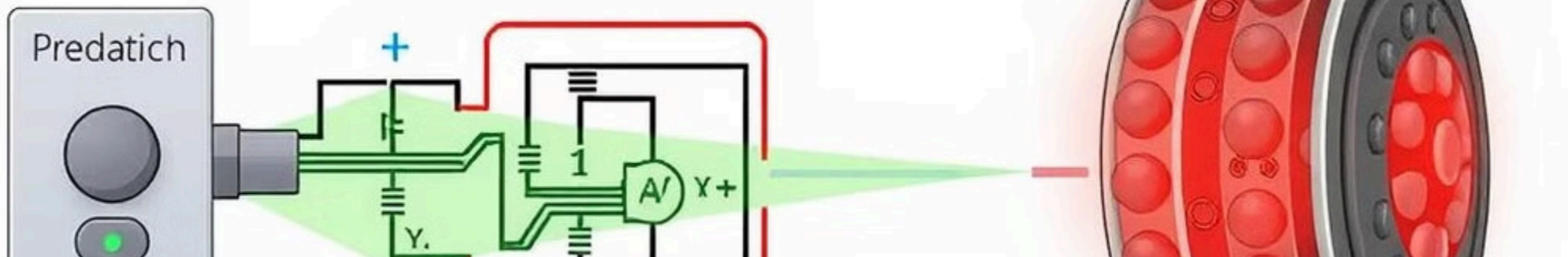
dcceeti.cirb)

Hesrorl.i+al

Sleaneer ind feeldling, erve bes.

Staling

Stall



Branch Prediction Techniques in MIPS

1

Static Prediction

Always predicts a branch to be taken or not taken, based on a predefined rule.

2

Dynamic Prediction

Uses the execution history of previous branches to predict the outcome of current branches.

3

Branch History Table

Stores recent branch outcomes, allowing for more accurate predictions based on historical patterns.

Exceptions and Interrupts in MIPS

1

Exceptions

Synchronous events that occur during program execution, such as an arithmetic overflow.

2

Interrupts

Asynchronous events that occur external to the CPU, such as a network packet arrival.

3

Exception Handling

A mechanism to detect, handle, and resume program execution after an exception or interrupt.

Advances in MIPS Microarchitecture

