Application Note

Zilog's eZ8™ CPU Versus Motorola's CPU08 - A Comparison Study

AN012302-0608

Abstract

Zilog's eZ8™ CPU is a high-performance 8-bit microcontroller Central Processor Unit (CPU) designed to address continuing demand for faster and more code-efficient microcontrollers. The eZ8 CPU supports a superset of the original Z8® instruction set and provides an upgrade path that includes compatibility and performance for Z8-based designs.

This Application Note compares the features and performance of the eZ8 CPU against the popular Motorola® CPU08.

eZ8 CPU Description

The eZ8 CPU is the central processor unit of Zilog's Z8 Encore!® family of microcontrollers. It is designed to meet the continuing demand for faster and more code-efficient microcontrollers. The features of the eZ8 CPU include:

- Direct register-to-register architecture that allows each register to function as an accumulator. This capability improves execution time and decreases the required Program Memory.
- Support for separate Program, Data and Register Address Spaces.
- Internal Register File that allows access of up to 4 KB.
- 20 MHz core and internal bus operation.
- Highly efficient instruction set that supports direct access to full register file. Also, support for software stack, bit manipulation, multiplication and multi-byte compare instructions to enable optimized implementation of higher-level programming language features.
- Pipelined instruction fetch and execution.

CPU08 CPU Description

The CPU08 is the CPU of Motorola's MC68HC908 family of microcontrollers. The CPU08 supports an accumulator based programming model with 64 KB of program/data space. The main features of the CPU08 include:

- 16-bit stack pointer with stack manipulation instructions.
- 16-bit index register (H:X) with high-byte and low-byte manipulation instructions.
- 8 MHz internal bus operation.
- 64 K of Program/Data Memory space.
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions.
- Low power STOP and WAIT modes.
- Pipelined instruction fetch and execution.

eZ8 Versus CPU08

We evaluate by performing a feature and performance comparison against the Motorola CPU08. The eZ8 versus CPU08 comparison is done under following 7 categories:

- Architecture
- Registers
- Addressing Modes
- Instruction Set
- High Level Language Support
- Resets and Interrupts
- Benchmark Programs

Table 1 lists the eZ8™ and CPU08 comparison details. A '+' in the last column of this table indicates features/measures where the eZ8 CPU is stronger. Below are the important highlights:

Architecture

The architecture, Address Space, and Program/Data Memory is provided below:

- With a modified Harvard Architecture, the eZ8 CPU is capable of supporting three different Address Spaces: 4 KB Register File, 64 KB Data Memory and 64 KB Program Memory.

  In comparison, the CPU08 supports a single 64 KB program/data space.

- The eZ8 CPU supports 20 MHz core and internal bus operations.

  In comparison, the CPU08 supports 8 MHz internal bus operation.

Registers

The eZ8 CPU supports uniform direct access to the entire 4 KB Register File. All the registers can function together as an Accumulator. The result is a flexible instruction set for programming and for compiler code generation.

In comparison, the CPU08 supports an Accumulator-based instruction set. A majority of the data movements and operations can happen only through the Accumulator, thus making the Accumulator a potential bottleneck.

Addressing Modes

- The eZ8 supports a very flexible register access scheme. Registers can be accessed as part of a Working Group (4-bit addressing), or as part of a Page (8-bit addressing), or directly (12-bit addressing) to improve execution time and reduce program space.

  In the CPU08, the first 256 Bytes of the 64 KB program/data space is treated as registers (8-bit addressing) for direct access.

- In the eZ8 CPU, with indirect addressing, the entire Data and Program spaces can be accessed with 16-bit register pairs. Indirect addressing is not supported in CPU08.

- The eZ8 CPU supports 8-bit indexed access for program and data memory loads.

  The CPU08 supports a flexible indexed addressing mode with 8-bit and 16-bit offsets or no offset. The Stack Pointer can also be used in place of the index register.

- Memory-to-memory data transfer capability is very critical in application performance. The eZ8 CPU supports direct data moves between locations and also initialization by immediate data in the entire 4 KB Register File.

  This capability is restricted to page 0 (first 256 Bytes) in the CPU08.

Instruction Set

- The entire Register File of the eZ8 CPU serves as multiple accumulators.

- In the eZ8 CPU, Push and Pop instructions are usable for the entire Register File. The Push and Pop instructions are restricted to the Accumulator and Index Register in the CPU08.

- The following instructions are supported in the eZ8 CPU only:
  - Bit Test Complement Under Mask
  - Bit Swap
  - Compare With Carry
  - Increment/Decrement Word
  - Jump Direct (using 16-bit address) on flags/condition codes
  - Rotate Right
  - Rotate Left

- In comparison, the following instructions are only supported by the CPU08:
  - 16-bit by 8-bit Divide
  - Compare_&_Branch_if_equal (CBEQ)
  - Negate (2's complement)
  - Logical Shift Left

High Level Language Support

To enable efficient compiler code generation, the eZ8™ CPU supports:

- Compare With Carry instructions for word and long word comparisons.
- Decrement and Increment word instructions for pointer arithmetic.
- Push/Pop support for the entire Register File.
- Direct Jump on flags/condition codes using 16-bit addresses.
- 8-bit by 8-bit multiplication.
- Decrement and Jump if Not Zero (DJNZ) instruction for loop counting.
- Data pointer testing (two instructions: Compare and Branch).
- Automatic save and restore of stack frame on interrupts.

In comparison, the CPU08 supports only the last four features listed above and a 16-bit by 8-bit divide.

Resets and Interrupts

- The eZ8 CPU supports up to 256 interrupts, hardware and software combined. Hardware interrupts can be level- or edge-sensitive with a three level programmable priority. Software interrupts are generated by the Trap instruction which supports up to 256 interrupt vectors.

  In comparison, the CPU08 supports up to 128 interrupts, only one of which is a software interrupt. Hardware interrupts are level- or edge-sensitive with a pre-assigned priority. A software interrupt is generated by the SWI instruction supporting only a single interrupt vector.

- The maximum interrupt latency, computed as the number of cycles from the interrupt occurence to the first ISR instruction execution, is 21 cycles (1.05 µs) for the eZ8 CPU.

  For the CPU08, the maximum interrupt latency is 20 cycles (2.5 µs).

Benchmark Programs

We have used general-purpose commonly used benchmark routines to compare the performance of the eZ8 and CPU08 CPUs. These are not routines made up to highlight any one instruction set. The benchmark programs exercise data movement, arithmetic operations, I/O manipulation and time keeping capabilities of the CPU. The eZ8 CPU is capable of providing a speed up of 3x or more in execution time utilizing the same or less code space.

For program description and code details, refer to the *Zilog® eZ8 CPU Performance Benchmarking Application Note (AN0125)*, available for download at www.zilog.com.

Table 1. eZ8™ vs CPU08 Comparisons

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
Architecture	Internal Bus Architecture	Modified Harvard Architecture with separate program and data memory bus.	Von Neumann Architecture with a single program/ data memory bus.	+
	Pipeline Structure	Fetch and Execution overlap.	Fetch and Execution overlap.	=
	Instruction Cycle Time	0.05 μs	0.125 μs	+
	Cycles/Instruction	2 to 9	1 to 9	=
	Internal Bus Frequency	20 MHz	8 MHz	+
	Performance	Up to 10 MIPS.	Up to 4 MIPS.	+
	Address Spaces	Separate Program, Data and Register Spaces.	Single Program/Data Space.	+
	Byte Ordering	Big Endian.	Big Endian.	=
	Instruction Set	Orthogonal Instruction Set – All registers can be used interchangeably.	Accumulator based Instruction Set.	+
	On-chip Memory	Up to 64 K Program and 4 K Register File.	Total of 64 K Program/ Data.	+
	Power Saving	Low Power Modes (Halt & Stop).	Low Power Modes (Wait & Stop).	=
	Peripheral Mapping	Memory Mapped.	Memory Mapped.	=
Registers	General-Purpose Registers (GPRs)	Full Register file (4 K) minus the upper 256 Bytes.	Only the first 256 Bytes of the memory map in direct access.	+
	Accumulator	All registers can be used interchangeably.	A dedicated Accumulator.	+
	Index Register	All registers can be used interchangeably.	16-bit Index Register.	+
	CPU Registers	Flags, Register Pointer and Stack Pointer.	Accumulator, Flags, Index and Stack Pointer.	=
	Register file/Data Memory (SRAM) Access	Flexible Indirect (16-bit), Extended (12-bit), Paged (8-bit), and Working Group (4-bit) to save code size.	Extended (16-bit) and Direct Page (8-bit).	+

Table 1. eZ8™ vs CPU08 Comparisons (Continued)

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
Registers (continued)	Program Counter	16-bit PC.	16-bit PC.	=
	Flag/Status Register	8-bit Flags.	8-bit Flags.	=
	Stack Pointer	16-bit Stack Pointer.	16-bit Stack Pointer.	=
Addressing Modes	Inherent Operand	Very Few (CCF, RCF, SCF).	Many accumulator based instructions are inherent operand.	=
	Register-4bit	Yes, saves code size	No	+
	Register-8bit	Yes	Yes	=
	Register-Extended	Yes, 12-bit extended mode.	Yes, 16-bit extended mode (one operand is accumulator always).	+
	Indirect	Yes, support for register, Program and Data Memory indirect access.	Not supported.	+
	Indexed	Yes, 8-bit signed offset. All registers can be used interchangeably.	Yes, supports a dedicated index register with 0, 8-bit, and 16-bit offsets. SP also could be indexed with 8-bit and 16-bit offsets.	-
	Relative	Yes, 8-bit signed offset conditional jumps.	Yes, 8-bit signed offset conditional branches.	=
	Direct	Yes, 16-bit direct address in jump and call instructions.	Yes, 16-bit address in jump and jump to subroutine instructions.	=
	Memory-to-Memory	Yes, (a) block transfer with auto-increment (b) register to register move in entire Register File.	Limited, (a) one operand in direct page and second operand through index register (b) register to register move oin page 0 only.	+
	Immediate	Yes, 8-bit immediate value.	Yes, 8-bit immediate value.	=
Instructions	Instruction Set	Orthogonal Instruction Set – All registers can be used interchangeably.	Accumulator based Instruction Set.	+

Table 1. eZ8™ vs CPU08 Comparisons (Continued)

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
Instructions (continued)	Bit Manipulation	Bit Set & Clear.	Bit Set & Clear.	=
		Carry Bit Set, Clear, & Complement.	Carry Bit Set & Clear.	=
		Jump if Bit is Set.	Branch if bit is set.	=
		Jump if Bit is Clear.	Branch if bit is clear.	=
		Bit Test Under Mask.	Bit test only for accumulator.	+
		Bit Test Complement Under Mask.	Not supported.	+
		Bit Swap & Nibble Swap.	Only Nibble Swap on accumulator.	+
	Arithmetic	Add with/without Carry.	Add with/without carry, one operand must be accumulator.	+
		Subtract with/without Carry.	Subtract with/without Carry, one operand must be accumulator.	+
		Increment Byte, Increment Word.	Increment Byte	+
		Decrement Byte, Decrement Word.	Decrement Byte.	+
		Compare.	Compare, one operand must be accumulator/index register.	+
		Compare with carry (for multi-byte compare).	Not supported.	+
		Decimal Adjust - works following addition and subtraction.	Decimal Adjust only for accumulator. Also, works only following an addition.	+
		8-bit Unsigned Multiply.	8-bit Unsigned Multiply with operands in accumulator and index register.	+
		Not supported.	16-bit by 8-bit divide.	-
	Logical	AND, OR, XOR	AND, OR, EOR: One operand must be accumulator	+

Table 1. eZ8™ vs CPU08 Comparisons (Continued)

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
Instructions (continued)	Logical	COM (One's complement).	COM and NEG (One's and Two's complement).	-
	Shift and Rotate	Shift Right Arithmetic.	Arithmetic Shift Right.	=
		Shift Right Logical.	Logical Shift Right.	=
		Rotate Left through Carry.	Rotate Left through Carry.	=
		Rotate Right through Carry.	Rotate Right through Carry.	=
		Rotate Right	-	+
		Rotate Left	-	+
		-	Arithmetic/Logical Shift Left	-
	Load	Load Memory Immediate.	Limited, load immediate only to first page.	+
		Load Register to/from Memory.	Load and Store only for accumulator and index register.	+
		Memory-to-memory block move with auto increment.	Limited, one operand in direct page and second operand through index register.	+
		Load to/from Memory indexed with 8-bit signed offset.	Indexed load/store only for accumulator using 8-bit and 16-bits and no offsets.	-
		Push/Pop for entire register file.	Push/Pop only for accumulator and index register.	+
	Program Control	Jump Relative on flags set/clear.	Branch Relative on Flags set/clear.	=
		Jump Relative on LT, LE, GT, GE.	Branch Relative on LT, LE, GT, GE.	=
		Jump Direct on Flags set/clear.	Not supported.	+
		Jump Direct on LT, LE, GT, GE.	Not supported.	+
		Jump Always (relative and direct).	Branch Always (relative and direct).	=
		Call Direct.	Call Relative and Call Direct.	-

Table 1. eZ8™ vs CPU08 Comparisons (Continued)

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
Instructions (continued)	Program Control	Return from subroutine and Return from Interrupt.	Return from subroutine and Return from Interrupt.	=
		Loop counting: Decrement and Jump if Not Zero (one instruction).	Loop counting: Decrement and Branch if Not Zero (one instruction).	=
		Data Pointer Testing: Compare and Jump If Equal (two instructions).	Data Pointer Testing: Compare and Branch If Equal (one instruction).	-
		Software Interrupt (Trap) instruction supporting a maximum of 256 possible vectors.	Software Interrupt (SWI) instruction supporting a single vector.	+
		Jump if interrupt register bit set/clear.	Branch if IRQ set/clear	=
			Branch if interrupt mask set/clear.	
		Jump if half carry bit in flag register set/clear.	Branch if half carry set/clear.	=
	Logical	AND, OR, XOR.	AND, OR, XOR: One operand must be accumulator.	+
		COM (One's complement).	COM and NEG (One's and two's complement).	-
	Shift and Rotate	Shift Right Arithmetic.	Arithmetic Shift Right.	=
		Shift Right Logical.	Logical Shift Right.	=
		Rotate Left through Carry.	Rotate Left through Carry.	=
		Rotate Right through Carry.	Rotate Right through Carry.	=
		Rotate Right.	Not supported.	+
		Rotate Left.	Not supported.	+
		Not supported.	Arithmetic/Logical Shift Left.	-

Table 1. eZ8™ vs CPU08 Comparisons (Continued)

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
High Level Language Support	Instruction Set	All registers can be used interchangeably - Saves code space.	Accumulator based instruction set.	+
	Address Space	Separate Code and Data space improves code generation and register allocation.	Single Program/Data space.	+
	Addressing Modes	Very flexible register and memory addressing modes.	Direct memory access for only 256 Bytes.	+
	Multi-byte Arithmetic	Yes, ADC and SBC instructions.	Yes.	=
	Multi-byte Increment/ Decrement	Yes, both byte and word Increment/Decrement	Only byte Increment/ Decrement.	+
	Multi-byte Compare	Yes, CPC instruction.	No.	+
	Loop Counting	Yes, single instruction.	Yes, single instruction.	=
	Data Pointer Testing	Yes, two instructions.	Yes, single instruction.	-
	Multiplication Support	Yes, using any register pair.	Yes, operands must be in A and X.	+
	Divide Support	No.	Yes.	-
	Block Move	Yes, block move instruction with auto increment.	Limited.	+
	Stack Frame Save and Restore	Auto save/restore for PC and Flags (3 bus cycles).	Auto save/restore for PC, Accumulator, IX (low) and Flags (5 bus cycles).	=
	16-bit Index	Yes.	Yes.	=
	16-bit Stack	Yes.	Yes.	=
	16-bit Conditional Branch	Yes.	No.	+
	8-bit Signed Conditional Branch	Yes.	Yes.	=
Resets and Interrupts	Vectored	Yes, up to 256 interrupt vectors (hardware and software together).	Yes, up to 128 interrupt vectors (hardware and software together).	+
	Priority	Yes, 3 level individually programmable priority.	Not programmable. Pre assigned priority.	+

Table 1. eZ8™ vs CPU08 Comparisons (Continued)

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
Resets and Interrupts (continued)	External Interrupts	12 GPIO port pin interrupt sources.	A dedicated IRQ pin and GPIO pins	+
	Software Interrupts	Yes, TRAP instruction supporting up to 256 vectored interrupts.	Yes, SWI instruction supporting one vectored interrupt.	+
	Mask & Acknowledge	Yes, individual interrupt Mask and Acknowledge.	Yes.	=
	Assertion	Level and Edge.	Level and Edge.	=
	Edge Sensitivity	Programmable, falling or rising.	Falling Edge only.	+
	Maximum Interrupt Latency (Prioritization + Recognition + Stacking + Vector Fetching+ Instruction Fetching)	16 cycles 0.8 µs	17 cycles 2.1 µs	+
		21 cycles {2 + 9 (worst case) + 3 + 2 + 5 (worst case)} 1.05 µs.	20 cycles {2 + 7 (worst case) + 5 + 2 + 4 (worst case)} 2.5 µs.	+
	Reset Sources	Power-On Reset.	Power-On Reset.	=
		Voltage Brownout (VBO).	Low Voltage Inhibit (LVI).	=
		WDT Timeout.	COP Timeout.	=
		External Reset Pin Assertion.	External Reset Pin.	=
		Illegal Opcode.	Illegal Opcode and Address.	=
Benchmarks* (optimized for speed)	Packing Binary Coded Decimal (BCD)	5 cycles 4 bytes 0.25 µs	12 cycles 7 bytes 1.5 µs	+
	Loop Control	3 cycles 2 bytes 0.15 µs	3 cycles 2 bytes 0.375 µs	+
	Bit Test & Branch	3 cycles 3 bytes 0.15 µs	5 cycles 3 bytes 0.625 µs	+
	Shifting out 8-bit Data & Clock	164 cycles 26 bytes 8.20 µs	205 cycles 25 bytes 25.625 µs	+

Table 1. eZ8™ vs CPU08 Comparisons (Continued)

Feature	Description	eZ8 CPU	Motorola CPU08	eZ8 Rating
Bench-marks* (optimized for speed)	Five Byte Block Move	14 cycles 15 bytes 0.7 µs	25 cycles 15 bytes 3.125 µs	+
	Four Byte Binary Addition	12 cycles 12 bytes 0.6 µs	51 cycles 11 bytes 6.375 µs	+
	Three Byte Table Search	59 cycles 44 bytes 2.95 µs	68 cycles 45 bytes 8.5 µs	+
	Four Byte Packed BCD Subtraction	101 cycles 40 bytes 5.05 µs	154 cycles 39 bytes 19.25 µs	+
	Input / Output Manipulation	29/17/19 cycles 38 bytes 1.45 µs/0.85 µs/0.95 µs	37/28/34 cycles 39 bytes 4.625 µs/3.5 µs/4.25 µs	+
Note: *: Benchmarks are in development and testing. These are initial results.				

Summary

This Application Note compares the features and performance of the eZ8 CPU against the popular Motorola® CPU08. The compared features include architecture, registers, addressing modes, instructions, high level language support, reset and interrupts, and speed.

References

The documents associated with eZ8 CPU, Z8 Encore! MCU, and CPU08 CPU are provided below:

- Zilog eZ8™ CPU User Manual (UM0128)
- Z8 Encore! XP® F64XX Series Product Specification (PS0199)
- Motorola CPU08 Reference Manual, CPU08RM/AD, Rev. 3, 2/2001
- Motorola MC68HC908AB32 Microcontroller Technical Data, Rev. 1.0

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer