a) Conventional Microprocessor System

MEM

Main Storage
(System Memory)

Address

Data Bus

CPU

Processor Core

I/O

Ports & Peripherals
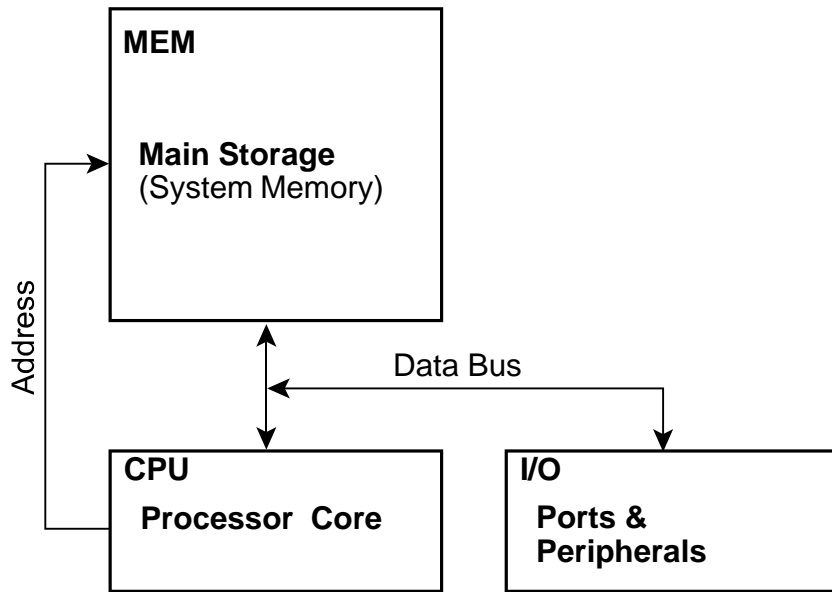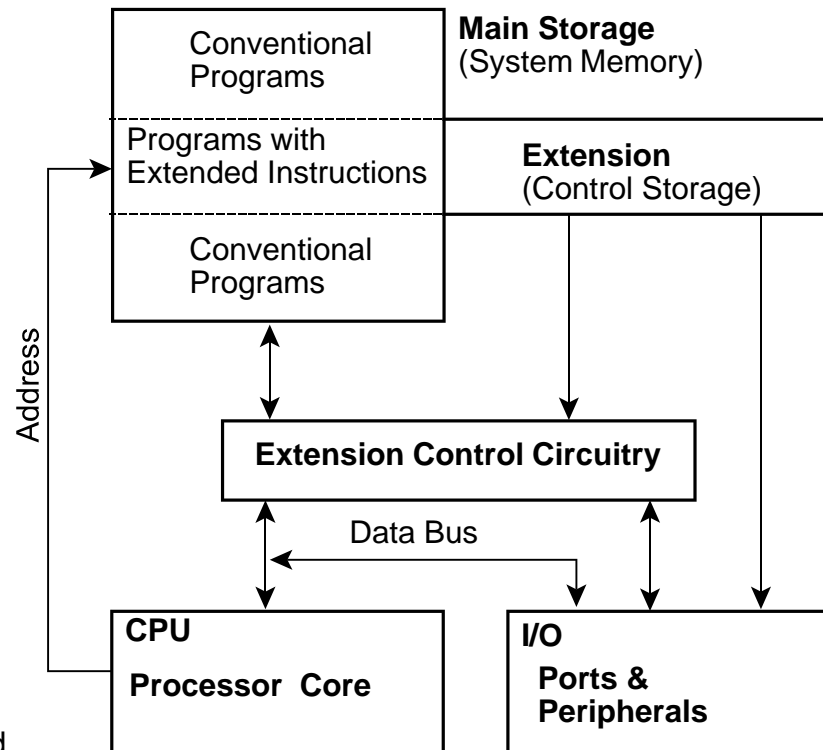
Extended instructions can be likened to horizontally encoded microinstructions. Add as many bits as required...

Extended and conventional instructions can be mixed freely. In contrast to a microprogrammed processor designed from scratch, the augmentation of the instruction set does not impede using existing software.

b) Microprocessor System with Extended Instructions

Conventional Programs

Main Storage
(System Memory)

Programs with Extended Instructions

Extension
(Control Storage)

Conventional Programs

Address

Extension Control Circuitry

Data Bus

CPU

Processor Core

I/O

Ports & Peripherals

Extending Instructions of Conventional Processors (1)
General Overview
11/4/2013

MEM

Main Storage
(System Memory)

Extension
(Control Storage)

A Conventional Instruction
in a Extended Memory Area

| OP | R1 | R2 | | 00 ... 00 | | |

An Extended Instruction

| OP | R1 | R2 | | CMD | CTL | EMIT |

A Conventional Instruction

| OP | R1 | R2 | etc. |
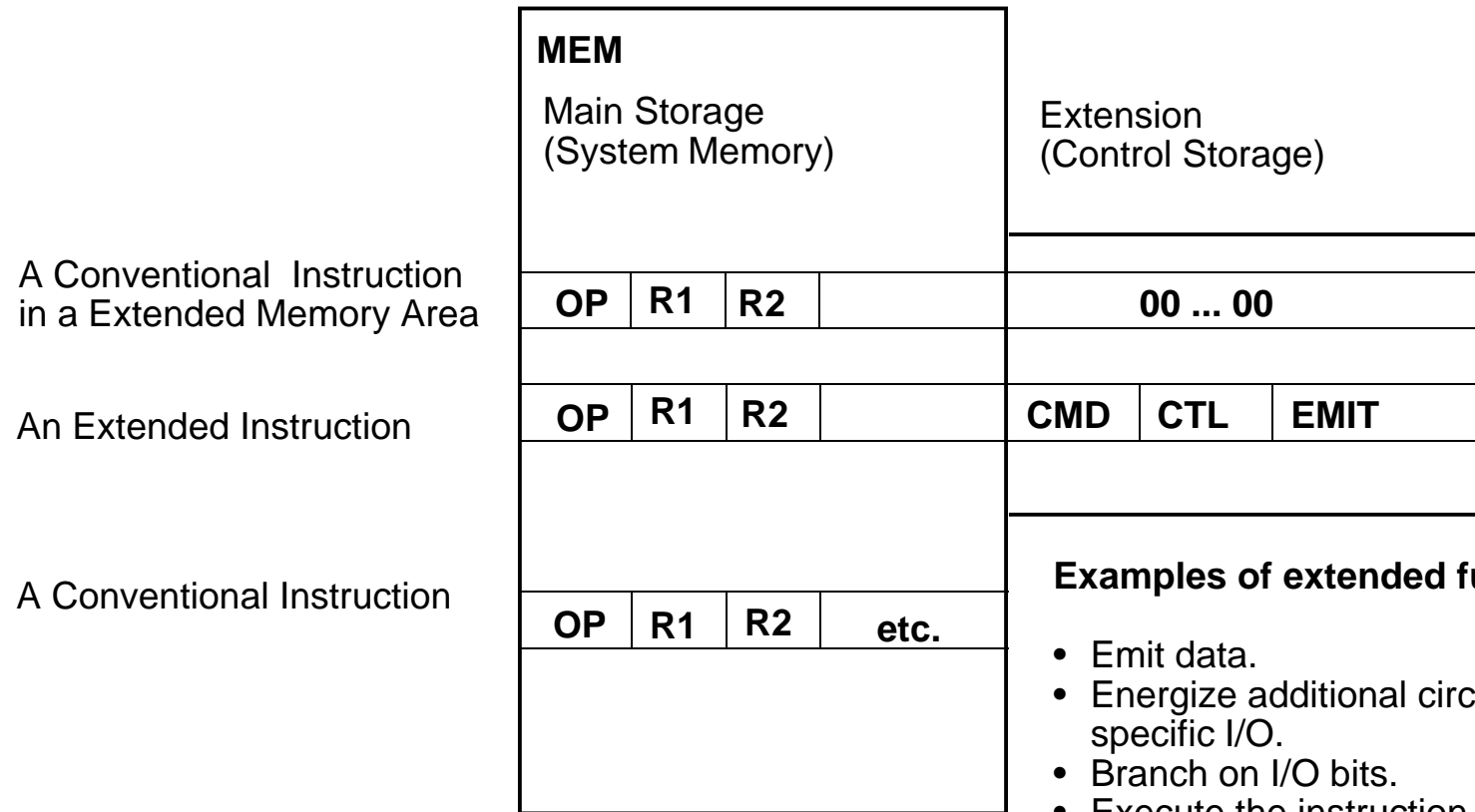
Examples of extended functions:

- Emit data.
- Energize additional circuitry, like accelerator units or application-specific I/O.
- Branch on I/O bits.
- Execute the instruction depending on certain conditions (predication).
- Execute an alternate instruction.
- Disable interrupts temporarily.
- Employ the address sent by the processor as additional data to be emitted.
- Facilitate compare stop functions (with an unlimited number of hardware breakpoints).
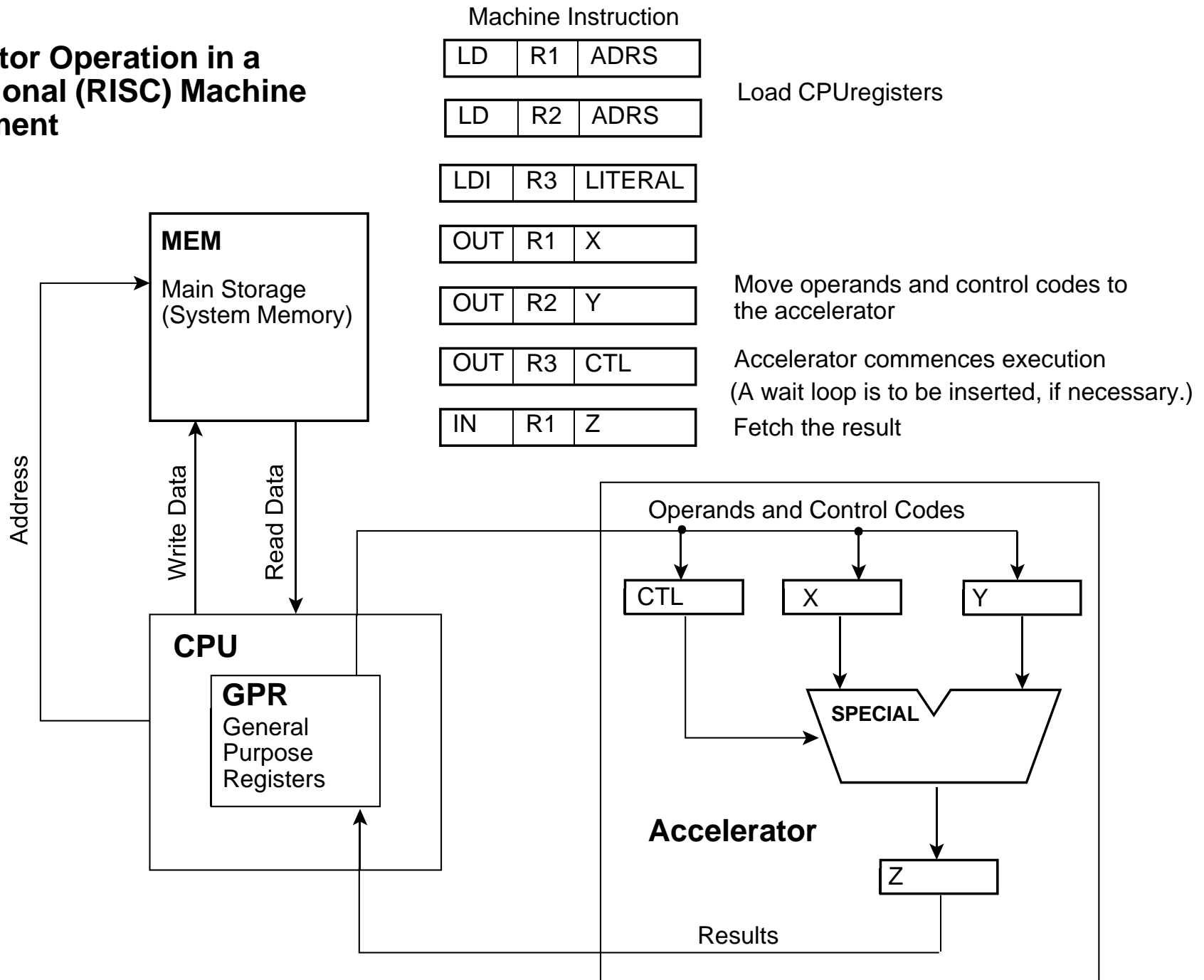
Extending Instructions of Conventional Processors (2)
Extension Principle
11/4/2013

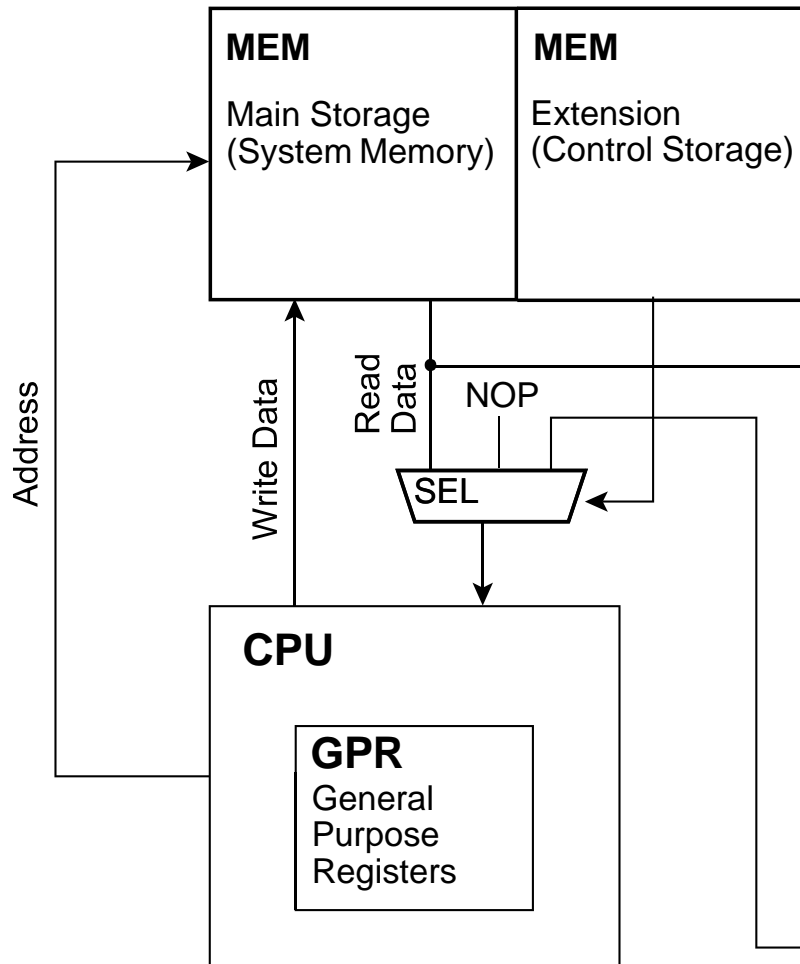# Accelerator Operation in a Conventional (RISC) Machine Environment

11/4/2013

Machine Instruction

| LD | R1 | ADRS |

| LD | R2 | ADRS |

Load CPU registers

| LDI | R3 | LITERAL |

| OUT | R1 | X |

| OUT | R2 | Y |

Move operands and control codes to the accelerator

| OUT | R3 | CTL |

Accelerator commences execution
(A wait loop is to be inserted, if necessary.)

| IN | R1 | Z |

Fetch the result

MEM

Main Storage
(System Memory)

Address

Write Data

Read Data

CPU

GPR
General Purpose Registers

Operands and Control Codes

| CTL | | X | | Y |

SPECIAL

Accelerator

| Z |

Results

Accelerator Operation
Supported by Extended Instructions
11/4/2013

Machine Instruction   **Extension**

LD	R1	ADRS		LOAD X
LD	R2	ADRS		LOAD Y
LDI	R3	LITERAL		LIT.TO CTL
LDI	R4	ADRS		Z to R4

} CPU receives a copy of data or a NOP

Accelerator commences execution*

CPU receives the result. ADRS is ignored.

*: Extended control means may insert wait states, if necessary.

MEM Main Storage (System Memory)

MEM Extension (Control Storage)

Address

Write Data

Read Data

NOP

SEL

CPU

GPR General Purpose Registers

Operands and Control Codes

CTL   X   Y

SPECIAL

Accelerator

Z

Results