HOCHSCHULE ESSLINGEN

LECTURE COMPUTER ARCHITECTURE

ADVANCED MICROPROCESSOR ARCHITECTURES: INTEL X86

RAINER KELLER

HOCHSCHULE
ESSLINGEN

CONTENT

1 Overview of Microarchitectures

2 Internals of Intel

3 Internals of ARM

GOALS FOR TODAY

- Overview of Architectures (CPU and MCU)
- Internals of Intel
- Internals of ARM

OVERVIEW 1/2

Microprocessors for General Purpose Computers (Standard PC) and Servers:

CPU Type	Supplier	Type/Data width	Comment
PCs, Workstations, Notebooks			
80x86, x86	Current: Intel, AMD, Zhaoxin	CISC, 32/64 Bit	Brand names: Intel Core, AMD Ryzen, … Internal names: Alder Lake, Meteor L. Lots of different manufacturers in the past
~~Power~~	~~IBM~~	~~RISC, 32/64 Bit~~	~~Until 2005 used in Apple Macs (later used in many laser-printers)~~
Server (file server, web server, database server)			
~~Itanium~~	~~Intel (HP)~~	~~RISC, 64 Bit~~	~~2001: Completely new 64-bit CPU, overtaken by AMD64 (x86-64),~~ **until 2021!**
Sparc	Sun/Oracle	RISC, 64 Bit	Oracle buys Sun in 2009, Licensing e.g. to Fujitsu as SPARC64 XII
Power	IBM	RISC, 64 Bit	Upward compatible to Power PC
Others:			
Mainframes	IBM/Fujitsu	RISC, 64 Bit	IBM zSeries and Fujitsu BS2000 allow financial services to run 5x9er: 99,999%
HPC	IBM, Cray/HPE, Dell, NEC, Fujitsu		Special purpose for HPC, but: no special CPUs, but rather “commodity” CPUs
GPUs	NVIDIA, AMD, Intel, others using FPGAs		Using graphics processing units for general-purpose: GPGPU…

Microprocessors for Mobile Devices

Game Consoles			
Power	IBM	RISC, 64 Bit	Microsoft XBOX 360, Nintendo Wii/Game Cube
~~Cell~~	~~IBM (Sony/Toshiba)~~	~~RISC, 64 Bit~~	~~Sony Playstation 3 (combination of PowerPC + 8-core Cell Broadband Engine)~~
x86	Intel, AMD	CISC, 64 Bit	Microsoft Xbox X, Sony Playstation 5
Smartphones, Tablets:			
ARM 9, ARM 11, ARM Cortex	Miscellaneous (Apple, Samsung, Qualcomm)	RISC, 32/64 Bit	Various different vendors, which license the Intellectual Property (IP) from ARM holding (now NVIDIA!), which include many different features (DSP, instruction extensions for Vectorization/SIMD/KI, FPGA)
RISC-V	Various	RISC, 32/64 Bit	Open Source CPU from University of Berkeley, very extensible, lots of research

OVERVIEW 2/2

Microcontrollers for Embedded Systems

CPU Type	Supplier	Type/Data width	Comment
Embedded Systems: Controllers			
8051	Misc. (Origin: Intel)	CISC, 8 bit	Licensed and enhanced by many suppliers, e.g. Philips/NXP, Siemens/Infineon
680x 681x 68xxx MPC55xx	Freescale/Motorola now NXP	CISC, 8 bit CISC, 16 bit CISC, 32 bit RISC, 32 bit	CPU families 6805, 6808, 6809 CPU families 6811, 6812, 6816 CPU families 68000 – 68060, 68331, ColdFire, … Embedded Power PC MPC555, 5556
PIC 1x PIC 24 AVR 8 AVR 32	Microchip (Atmel)	RISC, 8bit RISC, 16 bit RISC, 8 bit RISC, 32 bit	Families 12, 14, 16, 18 (= size of CPU address) ATtiny, Atmega
C16x TriCore TC1x	Infineon (Siemens)	RISC, 16 bit RISC, 32 bit	Used in many automotive ECUs
R8C, M16C R32C, SuperH 78Kxx, V850	Renesas (Joint venture of Hitachi, Mitsubishi and NEC)	CISC, 8/16 bit RISC, 32 bit	Many different product families from 4 bit to 32 bit, collection of the proprietary CPUs of the mother companies of the joint venture.
MIPS	Imagination (MIPS Technology)	RISC, 32/64 bit	Originally for workstations/server, today used in routers and set-top boxes, e.g. AVM Fritz Box (which uses multiple CPUs / MCUs)
ARM 7, ARM 9, ARM Cortex	Miscellaneous	RISC, 32 bit	Licensed to many suppliers, e.g. NXP/Freescale, Atmel, Philips, STM, Apple (!), Microsoft(!), …

Very broad market, dominated by many “old” architectures (8051, 68xx developed in the 70s and early 80s): very large range of CPU performance, memory size and peripherals, however strong growth and convergence toward ARM!

POWER SAVING MECHANISMS

Purpose: Reduce cooling effort, **increase** battery operating time!

Idea: Electrical power loss for CMOS logic $P \approx U^2 * f$
But: CMOS transistors switch faster at higher gate voltages
→ lower supply voltage U only possible at lower clock *f*

Measures: Reduce supply voltage *U* and clock frequency *f*, if no/low computing performance is required:
→ Programmable PLL for clock generator *f*
→ Programmable voltage supply *U*

Switching occurs in steps:
→ Switch off peripherals
→ Switch off (part of) units or the bus system
→ Switch off CPU→ reactivated periodically by timer interrupts if required by an interrupt of a peripheral
→ Use a lower-power Core (Big.LITTLE-Core design…)

CPU SUPPORT FOR OPERATING SYSTEMS 1/4

Protection and Monitoring:
Privilege Levels and Privileged Instructions

Purpose: Operating system shall monitor user programs and check their resource usage (CPU time, memory)

Idea: OS monitoring periodically activated by timer interrupt

Measures: CPU switches between two privilege levels:

→ Kernel/System mode (Operating System)
→ User mode (Application programs)

Critical instructions may only be executed in kernel mode:

- Enable/disable interrupts, i.e. user programs cannot disable interrupts (and thus can’t stop OS monitoring)
- Write (and read) peripheral registers and other management structures, i.e. the interrupt vector table
- User programs may only call operating system services via a protected mechanism, i.e. software interrupts

CPU SUPPORT FOR OPERATING SYSTEMS 2/4

Run-time Monitoring

Purpose:	CPU shall be stopped/restarted, if the CPU “hangs”/runs wild due to a soft- ware bug.
Idea:	Check that a task is executed correctly and periodically in time
Measures:	Hardware timer (**Watchdog**): written periodically, i.e. by writing certain data values in a register. If the watchdog is not written written correctly / in time, the watchdog will reset the CPU.

Detection of Memory Errors:

Purpose:	Detection of “flipped” memory bits (reliability problem of large Dynamic RAMs or Flash ROMs at high temperatures)
Idea:	Store redundant parity bits/check sums (CRC), generated by a HW unit when writing data and check when reading. Stop program on error or automatic error correction (ECC).

Detection of CPU HWs Errors:

Idea:	Program executed on two identical CPUs clock by clock (Lockstep CPU) → very costly, used in safety critical systems
Measures:	Results of ALU operations etc. monitored by hardware comparators, reset the CPUs if results do not match

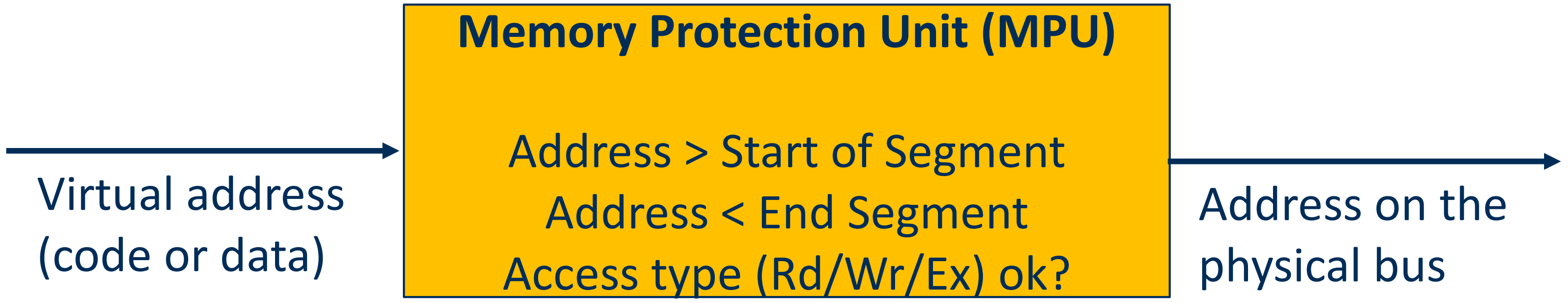
CPU SUPPORT FOR OPERATING SYSTEMS 3/4

Memory Protection (Segmentation)

Purpose: A user program must not access memory of the OS or of any other user program.

Idea: Each program gets its an exclusive memory address range (Segment)

Measures: HW monitoring and triggering interrupts, if a program tries to access memory (code or data) in a foreign memory address range.


Memory Protection Unit (MPU)
Address > Start of Segment
Address < End Segment
Access type (Rd/Wr/Ex) ok?
Virtual address (code or data)
Address on the physical bus

Example Freescale HCS12X

- 2 operating modes (supervisor and user state)
- 8 memory segments with configurable start and end addresses plus access rights (read, write and/or execute)

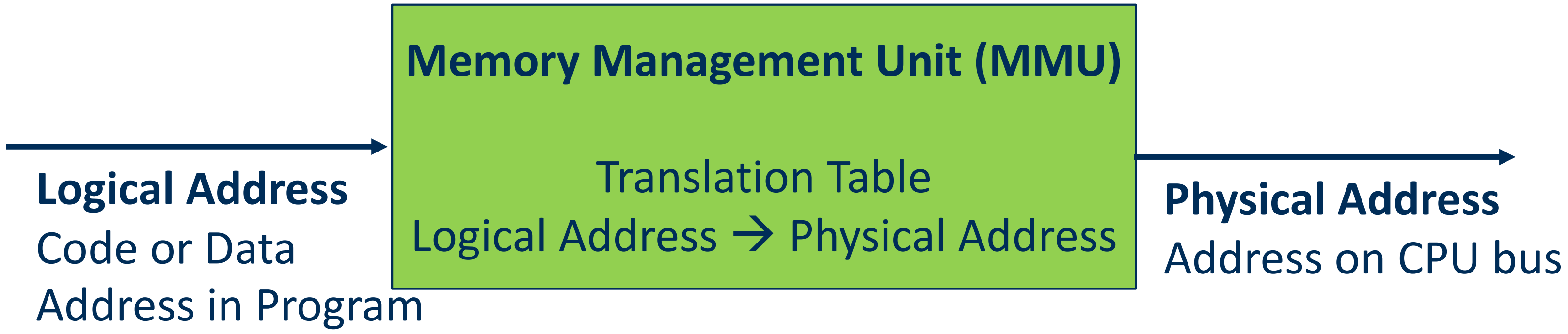
CPU SUPPORT FOR OPERATING SYSTEMS 4/4

Memory Extension (Virtual Memory / Paging / Swapping)

Purpose: Programs shall be able to use more memory than physically available or more, than the CPU’s address range allows.

Idea: If the memory is larger than the CPU’s address range (i.e. HCS12 DP256 with 256KB memory despite of 16bit addresses only):
Switching of memory blocks (Pages) via programmable address decoders
If more memory required than physically available: Extend the memory by copying currently unused data/code to the hard disk (Swapping)

Measures: HW monitoring and triggering interrupts, if a program tries to access memory (code or data) in a foreign memory address range.

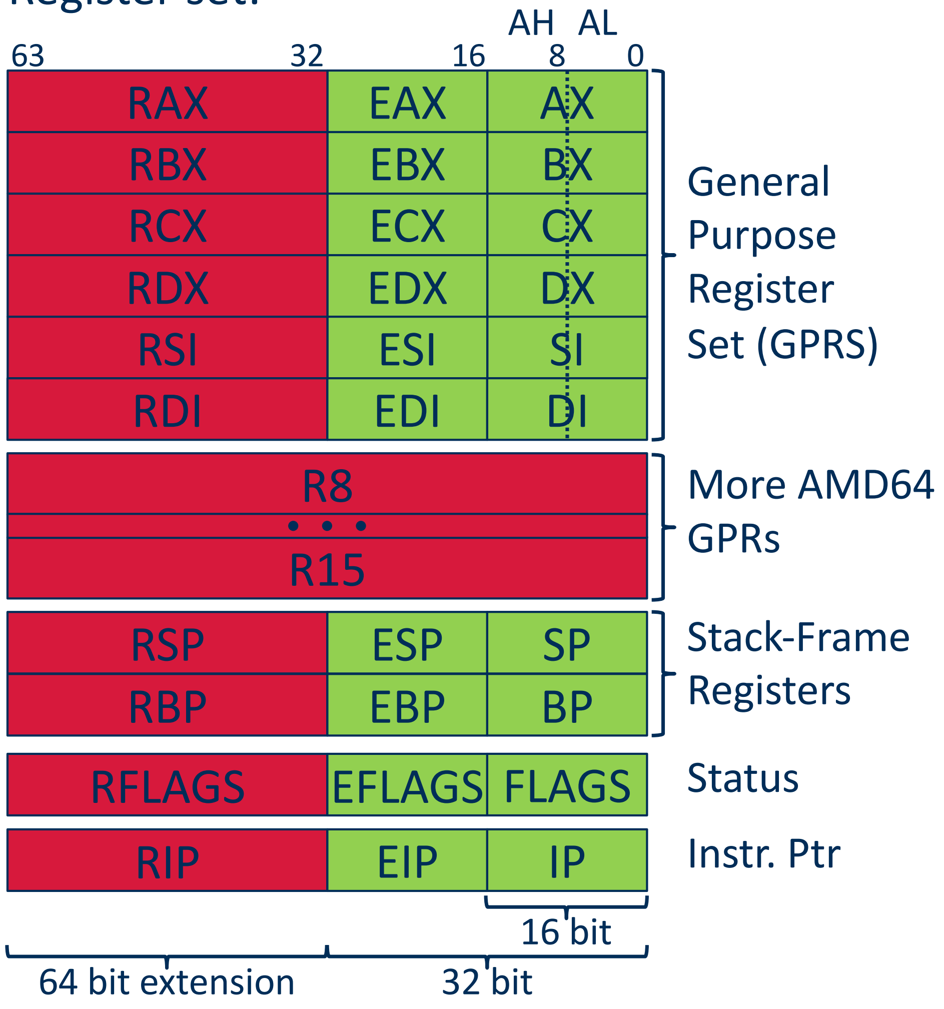


INTEL 80X86, X86 ARCHITECTURES

HOCHSCHULE ESSLINGEN

Programming Model:

Register set:



The 8 Floating pointer registers, MMX, SSE and AVX registers (s. next Slide)…

History:

- 8 Bit Arch.: 8080, 8085 (1974)
- 16 Bit Arch.: 8086 (1978), 80186, 80286
- 32 Bit Arch.: 80386 (1985), ‘486, Pentium
- 64 Bit Arch.: AMD64, EM64t, Core i3/i5/i7

Modes and OS:

- Real mode 16-bit DOS (IBM PC 1981)
- Protected mode 32-bit Windows (Win32)
- Compatibility mode 16/32bit (Win16)
- 64 bit “long” mode (Linux)
- Compatibility mode 32/64 bit (Win64)
- **FUTURE (2025?)**: X86s (64-bit only!)

Characteristics:

- Small register (only 6 GPRS reg.)
- **CISC** (>> 100 instruction)
- Up to **15 Byte** instruction length!
- **Von-Neumann** memory model
- **Little-Endian**, 1-byte addressable
- Address and data size 32/64 bit
- Mostly 2 address instructions: 1 reg+ 1 reg / memory operand

INTEL 80X86: ASSEMBLER

Example of 80x86 assembler instructions:

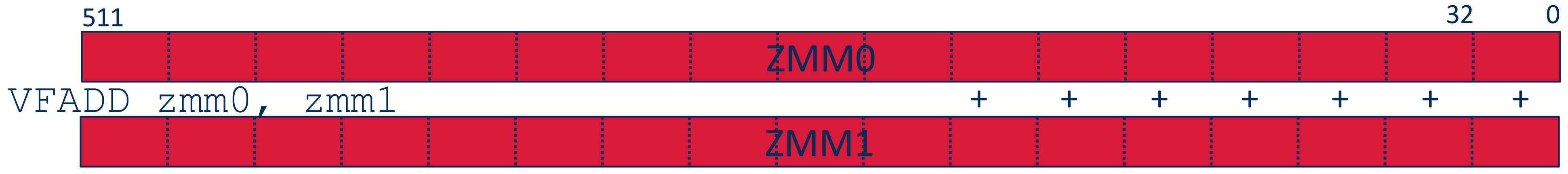
```
.data
myVar DD 01234567h, 89ABCDEFh

.code
MOV ESI, 4
MOV EAX, 1
ADD EAX, myVar[ESI]
```

Disassemble program (in Linux) using `objdump -d`

Possible addressing modes:

- Register addressing
- Immediate addressing
- direct, register-indirect and indexed, base-indexed memory addressing
- Floating point in 32-bit (Single Precision), 64-Bit and 80 bit (**legacy**) operands
- Single-Instruction Multiple Data (SIMD) with up to 16x 32-bit FP in AVX-512:



INTEL 80X86: INSTRUCTION SET

HOCHSCHULE ESSLINGEN

Orthogonal instruction set:

- Some registers have specific meanings: AX = Accumulator, CX=counter
- Most registers can be source and/or destination in (almost) all instructions

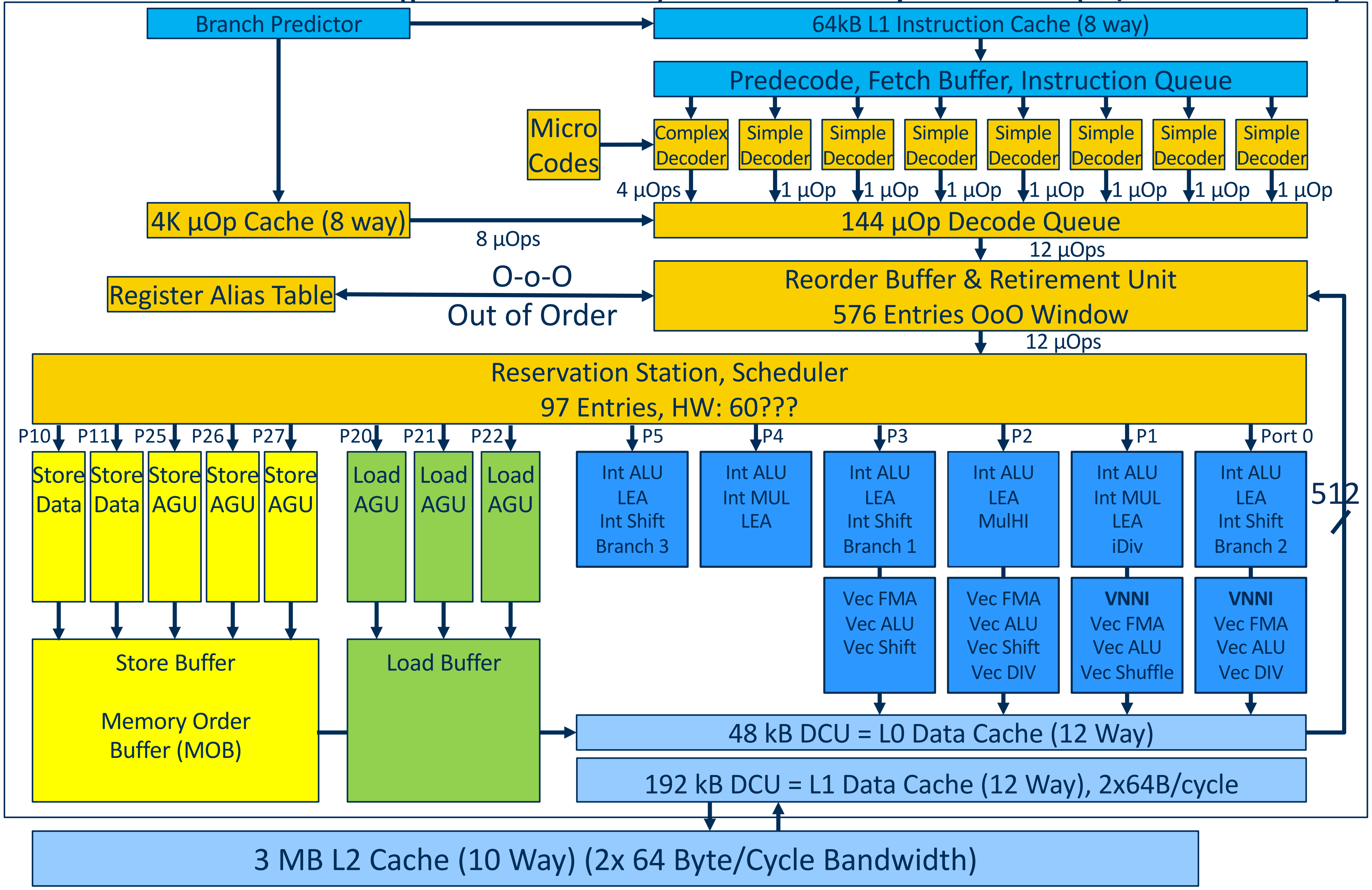
Major diff. to HCS12	Intel	Motorola/Freescale HCS12
Memory order	Little-Endian	Big-Endian
Operand order	Destination – Source `MOV EBX,EAX; EBX`$\leftarrow$`EAX`	Source – Destination `TFR A, B; A`$\rightarrow$`B`
Constants	`MOV EAX, 1`	`LDAA #1`
Hexadecimal values	`012345678h`	`$01234567`

Even though the register set and instruction set is ~40yrs old

- While external memory view von-Neumann, internal CPU caches is Harvard!
- CISC instructions are internally translated into µ-Ops (RISC)
- Superscalar Execution with multiple issues (see next slide) with SIMD!
- Medium- to long Instr. Pipeline (16 – 23 stages) with out-of-order exec., shadow registers, very powerful speculative branch prediction and prefetch!
- Dynamic clock switch and turning off units not in use.
- Big.LITTLE design beginning with Alder Lake (since 2021; ARM since 2012!)

INTEL ARROW LAKE (HERE LION COVE CORE), 10/2024

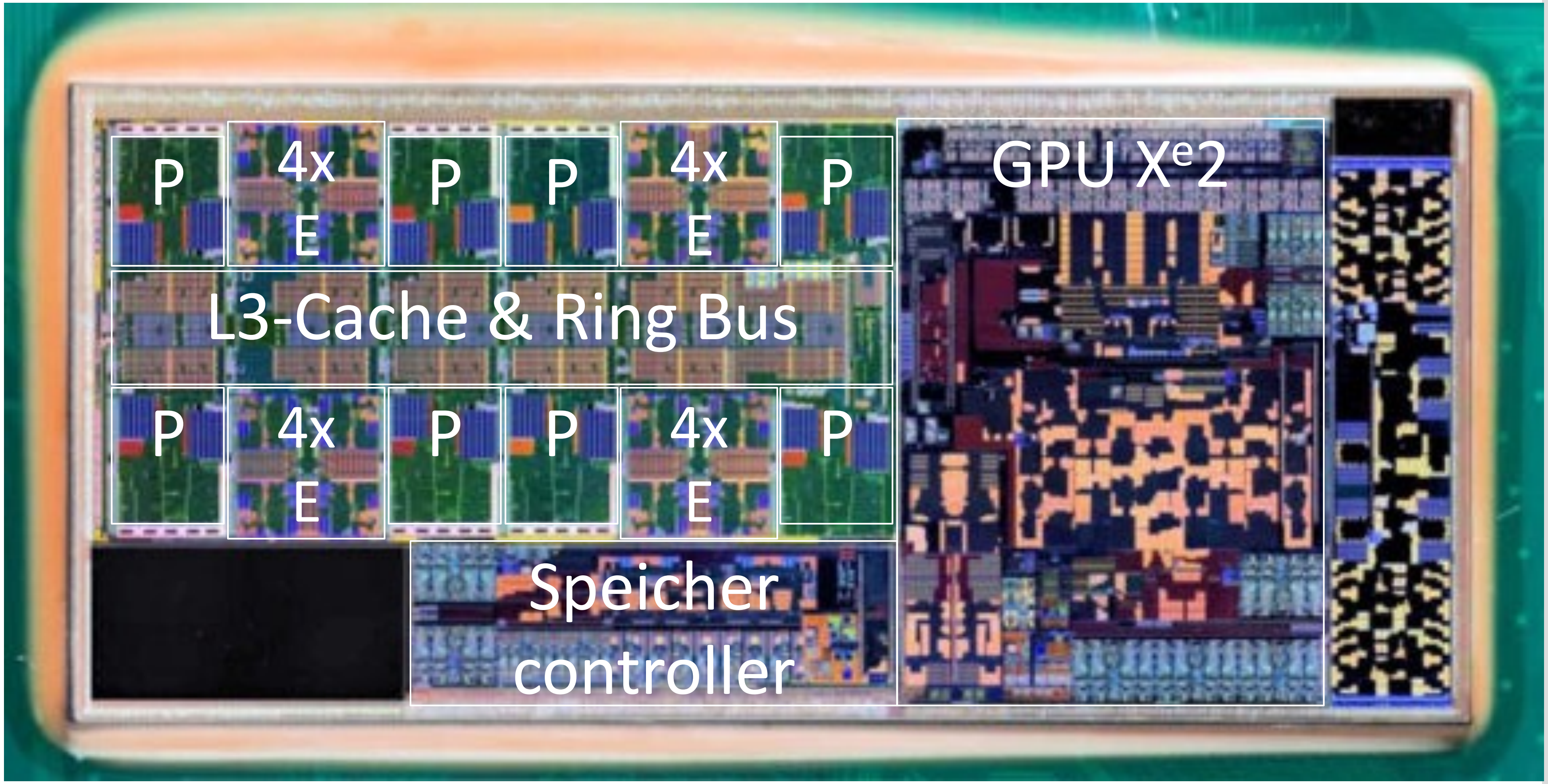
Performance P-cores (pictured here) vs. Efficiency E-Cores (w/o AVX et al)



INTEL ARROW LAKE, 10/2024

E-cores: Efficient cores

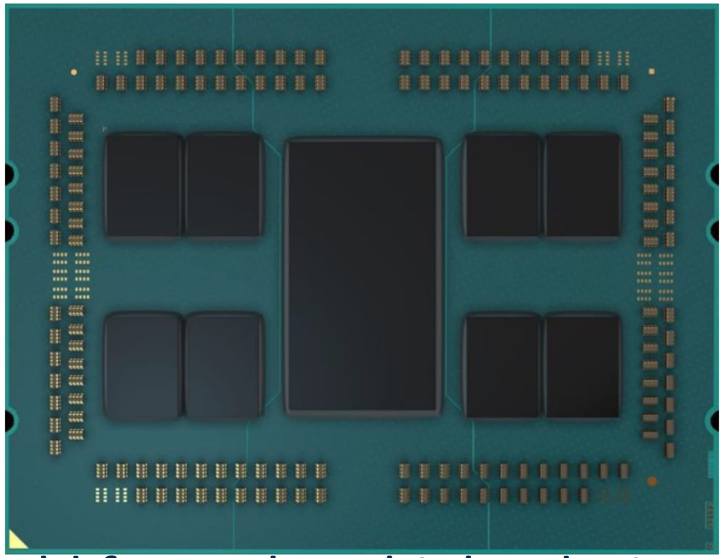
E.g. Arrow Lake with 8 P-cores and 16 E-cores (with a “Thread Director”):



See article Christian Hirsch: “**Das Imperium schlägt zurück**”, c’t p.84ff, issue 25, 2021 et al

AMD RYZEN

AMD’s Zen* architectures have proven to be very successful in servers!
Zen5 is produced with very advanced TSMC 4nm process for CCD, as well as TSMC’s 6nm for the Input/Output Die (IOD)



The yield for such a chiplet design is **good**™, aka cheaper CPUs!

Core Chiplet Die (CCD) contains up to 16 cores (1 CCD = 2 CCX á 8 cores)
CCD are interconnected via Infinity-Fabric

Zen 5-Turin: up to 128 cores per socket, with up to 512MB(!) L3 Cache with 12 channels into DDR5-6400MHz