



National
College of
Ireland



Hardware Intro

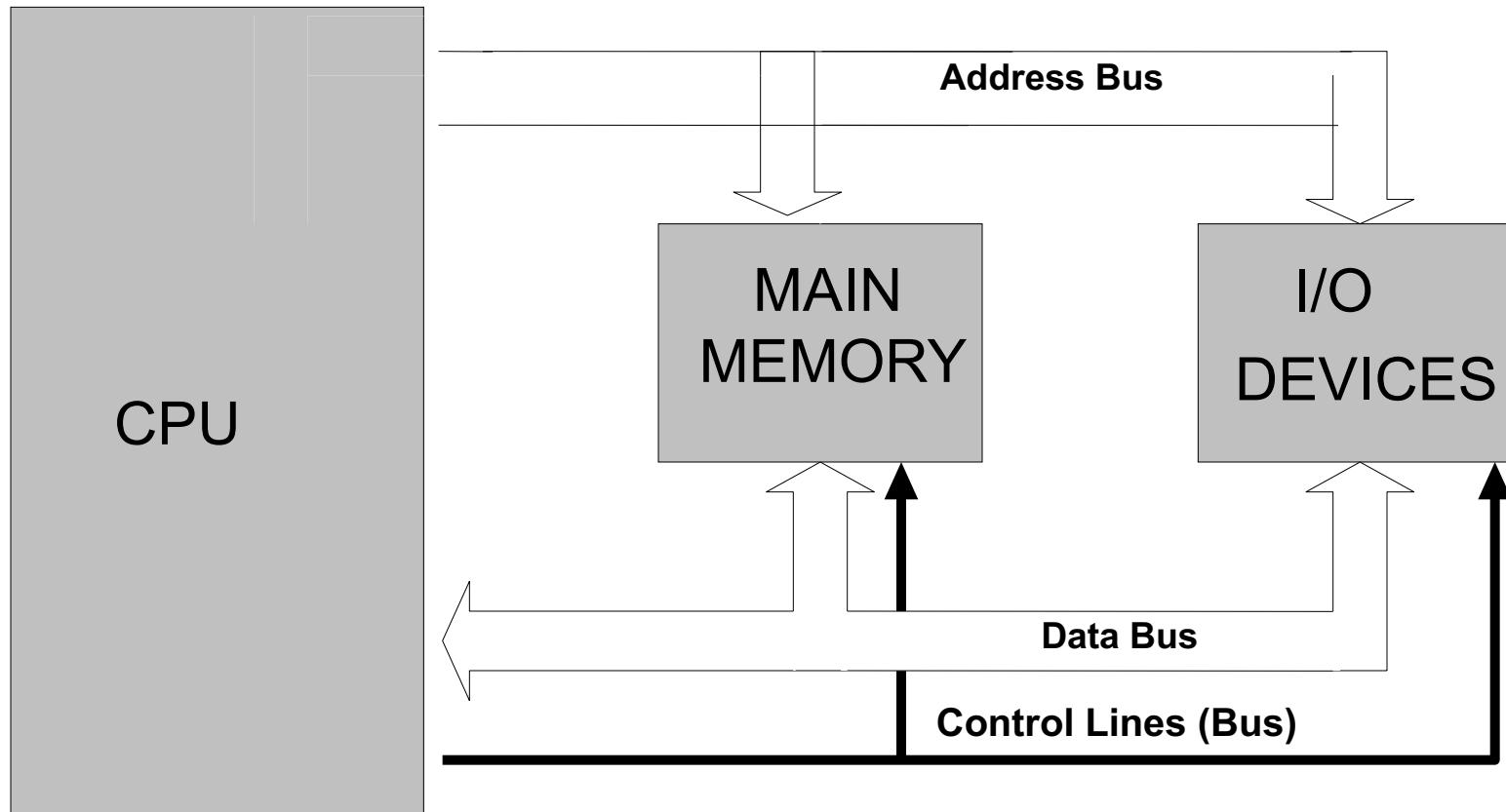
Courtesy of

Dr David Tracey (CLAR Evening Lecturer)

Course

- Revised Content
- 50% CA with supporting tutorial
- 50% Terminal Assessment (MCQ)

Simplified Microcomputer



Fetch Decode Execute

- Address of next instruction from PC (or IP in X86) is sent to memory on Address Bus
- Instruction (Op Code) is read into IR register from Data Bus for decoding
- Instruction is executed

Special Purpose Registers

- Program Counter (PC!) holds next instruction to be executed (Called Instruction Pointer in Intel x86 – Intel have a Code Segment too)
- Stack Pointer (SP) keeps track of the next location available on the Stack
 - STACK: part of main memory where program data can be stored (Used for flags, parameters). PUSH & POP operation – it is FILO
- Status or Flag Register (PUSH'ed onto STACK on function call along with parameters)
 - has bits for overflow, interrupt, carry, parity

Computer Systems

- A Computer performs millions of operations every second
 - System processor is executing instructions, data is being transferred from the system memory, storage devices are being read and written, input and output devices are sending and receiving information
- All this activity must be coordinated and managed.
- Computer's use special control signals and clock signals that synchronize components and set the pace for most internal operations.

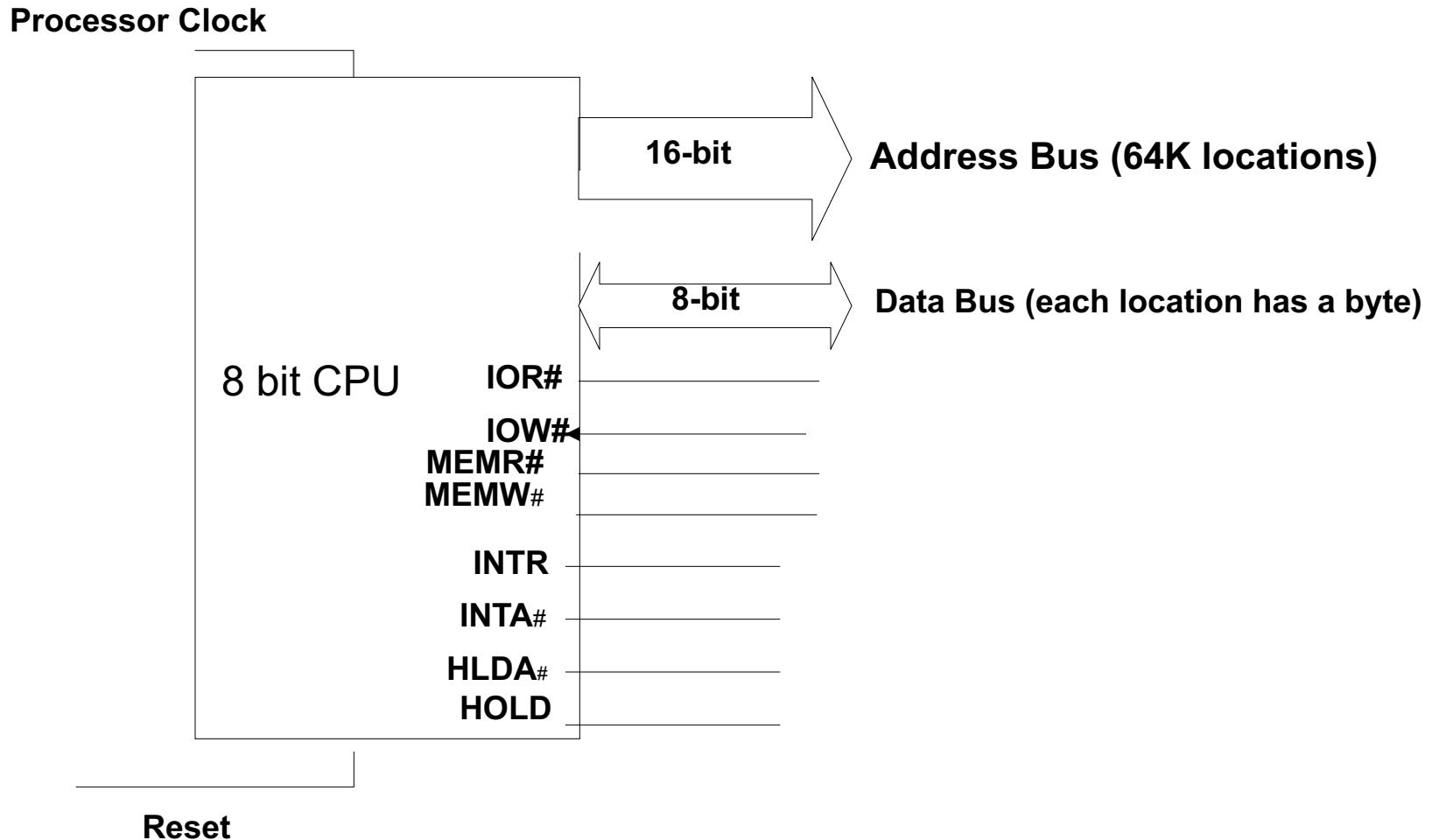
Signals

- A signal can be any type of information
 - data, address information, or control information or power
- Data bits are conceptually ones and zeros. Different components represent bits differently (hard disk - ones and zeros are encoded magnetically, PC circuitry– by voltage levels)
- Voltage is an electrical potential difference. 1 is generally + voltage, 0 is 0V
 - All components agree on the levels
 - the positive voltage must be sufficiently high that there's virtually no risk of a "one" being seen as a "zero" (or vice-versa).
- Computer hardware operates by looking for particular patterns in signals, and then responding to them
 - A memory chip recognizes a request to read a particular piece of data by looking for a particular value on a control line; when it senses that value, it looks at other signals for the address of the data to be read, and then responds by producing the requested data on a different set of signal lines.

CPU Signals

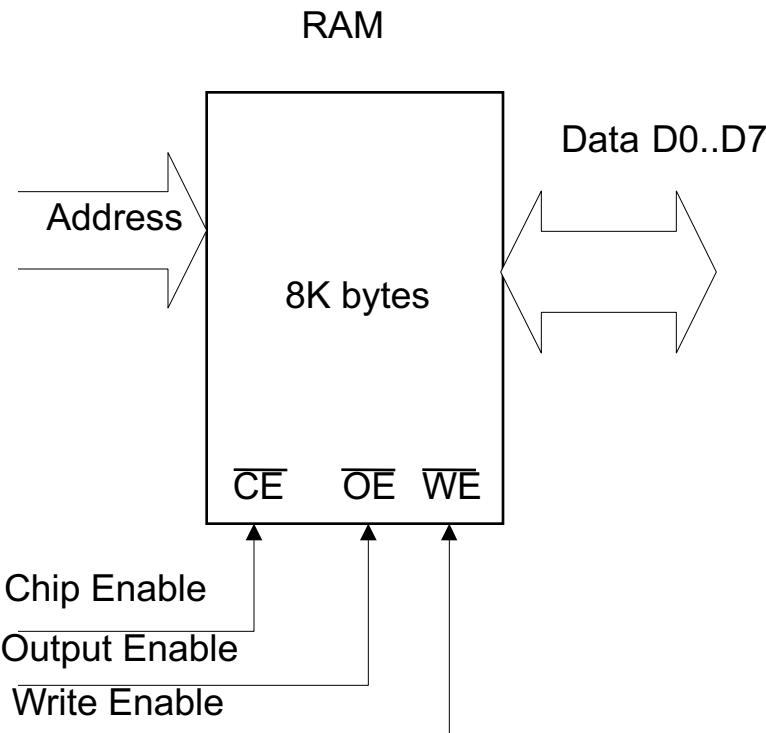
- Address Bus is Unidirectional
(set by CPU)
- Data Bus is Bidirectional
- Control Signals
 - Interrupt (INTR and INTA)
 - DMA (HOLD and HOLDA)
 - I/O Read/Write (IOR, IOW)
 - Memory Read/Write (MEMR, MEMW)

Simplified Microprocessor



Read Cycle

1. Processor puts out address on the Address Bus
 2. Processor asserts CE#
 3. Processor asserts MEMR# or OE
 4. Processor reads contents of the data bus
- Write is similar, but asserts MEMW# or WE

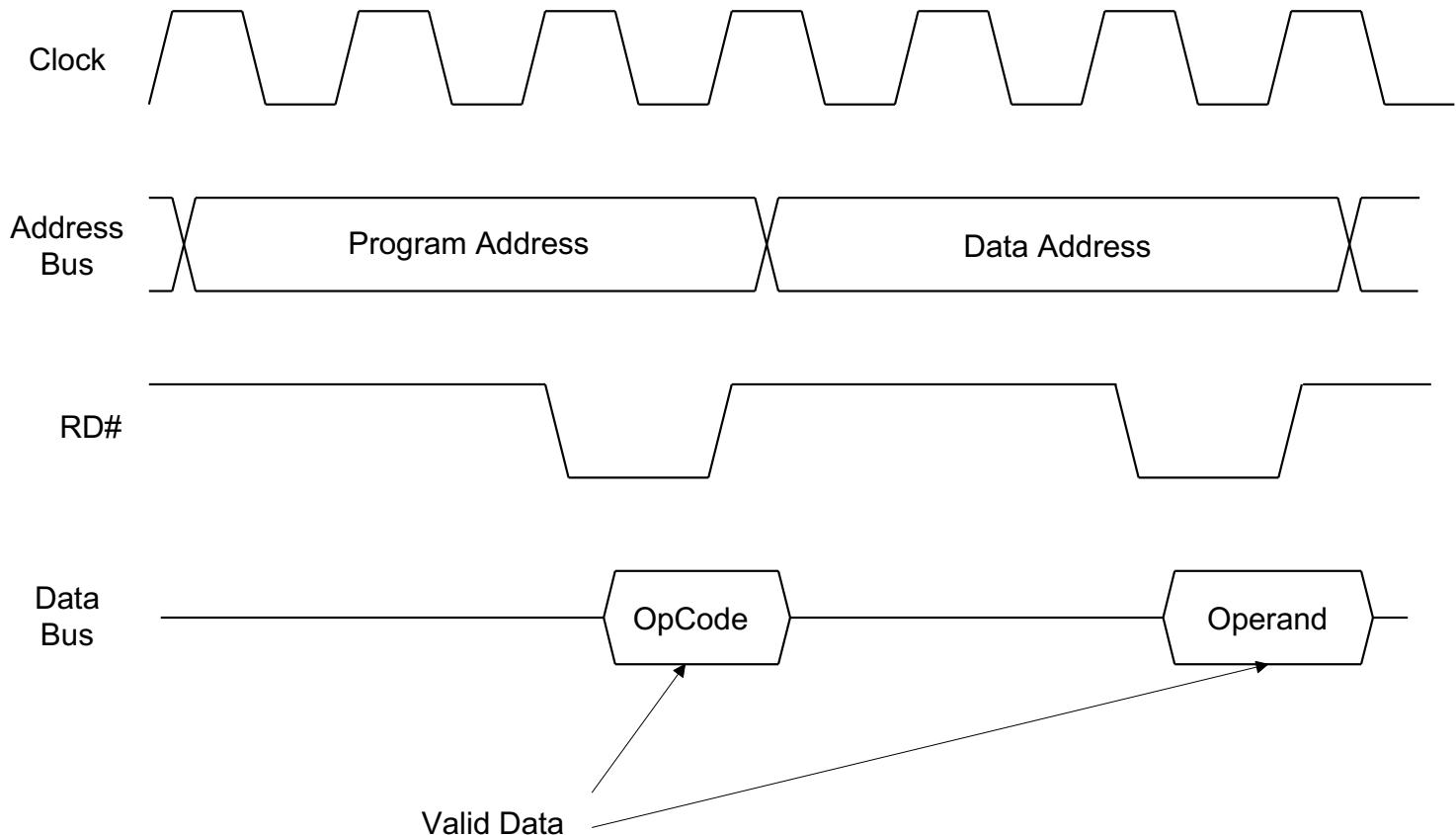


Memory Read Write Cycles

- Memory Chips have lines to enable and for output (read), write
 - CE (Chip Enable), OE (Output Enable), WE (Write Enable)
- CPU uses Control Lines for Memory Reads and Writes
 - Active low RD# asserted for a Read Cycle
 - Active low WR# indicates a write

Read Cycle Timing Diagram

(note CE happens before RD/MEMR asserted
memory cycle time is from when MEMR asserted to when it is off)



Direct Memory Access (DMA)

- Processor acts as middleman in normal transfer – can be a bottleneck
- DMA techniques improve system performance
 - used when very high data rates required
- Bus Master devices take control of the bus and do the work themselves
- Other methods (e.g. interrupts) use software to transfer data and are slower
- In theory, the processor can do other work simultaneously when DMA underway

Direct Memory Access

- In order to do bus mastering properly, a facility to arbitrate between requests to "take over the bus" must exist; this is provided by the chipset.
- Bus mastering is also called "first party" DMA since the work is controlled by the device doing the transfer.
- External devices can transfer data directly to or from memory under hardware control
- DMA Controller has signals like
 - IOR/IOW and MEMR/MEMW
 - HOLD to CPU and HOLDA from CPU
 - DREQ from (I/O) Device and DACK to device

Synchronous (Clocked) Data Transfer

- One of the most important functions of the clock signal is to control the transfer of data over an interface or bus. This is called *synchronous* or *clocked* data transfer.
- Most interfaces involve
 - one or more data signals that run between devices
 - control signals - tell various devices on the interface when to begin sending data, and when to look for data being sent by other devices. They also facilitate *negotiation*, which is the process of determining whose turn it is to use a system bus.
- Once a data transfer is ready to occur, the clock related to that interface or bus controls the transfer of each piece of data. In conventional operation, one bit of data is transferred across each data line, for each cycle of the clock.
- The "ticking of the clock" is recognized by triggering on either the rising edge or falling edge of the clock signal. Each subsequent rising or falling edge of the clock triggers the next chunk of data to move across the data line(s) from the sending to the receiving device.
- Since the pace of the clock controls the transfer of data, this means that the speed of the clock is also the speed of the bus or interface. Speeding up the clock means that data is transferred more quickly. The total throughput of any bus or interface is equal to the speed of the interface multiplied by the width of the data bus (how many data signals transfer data at once.)

Input and Output Cycles

- Some processors support a single address space for I/O devices and memory
 - Fewer instructions, more decode hardware
 - Fewer pins on chip
 - Specific I/O address block reserved
- Intel Architecture has separate I/O and memory address spaces
 - I/O devices decoded separately from memory devices
 - Specific IOR# and IOW# signals and instructions (e.g. IN – simpler, faster)

Voltage Levels and Signalling

- All signals have a name, e.g A11 for address line (bit) 11 and D2 for data bit 2.
 - /RESET represents a signal whose logic is inverted. It is true/on when it is zero, and false/off when it is one. So this means that this RESET signal is normally a one; when it drops to zero a reset occurs. This signal is **active low**
 - Signals change over time, except ground (zero) and power (one) signals. Devices respond to a signal by :
 - **Level-Triggered Activation:** device looks for a particular level on the signal, either a zero or a one and acts on seeing that level. Signals are normally active high OR
 - **Edge-Triggered Activation:** The device does not look for the level of the signal at all, but rather the *transition* from one level to another (either on both rise and fall or on both).
 - transition zero to a one is called the *rising edge*
 - transition from one to zero the *falling edge*.
 - Time-critical actions are more easily edge triggered, as transitions are often easier to detect and happen quickly, which allows for synchronized activity within the system
 - Levels are mainly used for data and regular control signals, and transitions for clock signals.

Year	Name	Made by	Comments
1834	Analytical Engine	Babbage	First attempt to build a digital computer
1936	Z1	Zuse	First working relay calculating machine
1943	COLOSSUS	British gov't	First electronic computer
1944	Mark I	Aiken	First American general-purpose computer
1946	ENIAC I	Eckert/Mauchley	Modern computer history starts here
1949	EDSAC	Wilkes	First stored-program computer
1951	Whirlwind I	M.I.T.	First real-time computer
1952	IAS	Von Neumann	Most current machines use this design
1960	PDP-1	DEC	First minicomputer (50 sold)
1961	1401	IBM	Enormously popular small business machine
1962	7094	IBM	Dominated scientific computing in the early 1960s
1963	B5000	Burroughs	First machine designed for a high-level language
1964	360	IBM	First product line designed as a family

1965	PDP-8	DEC	First mass-market minicomputer (50,000 sold)
1970	PDP-11	DEC	Dominated minicomputers in the 1970s
1974	8080	Intel	First general-purpose 8-bit computer on a chip
1974	CRAY-1	Cray	First vector supercomputer
1978	VAX	DEC	First 32-bit superminicomputer
1981	IBM PC	IBM	Started the modern personal computer era
1981	Osborne-1	Osborne	First portable computer
1983	Lisa	Apple	First personal computer with a GUI
1985	386	Intel	First 32-bit ancestor of the Pentium line
1985	MIPS	MIPS	First commercial RISC machine
1987	SPARC	Sun	First SPARC-based RISC workstation
1990	RS6000	IBM	First superscalar machine
1992	Alpha	DEC	First 64-bit personal computer
1993	Newton	Apple	First palmtop computer

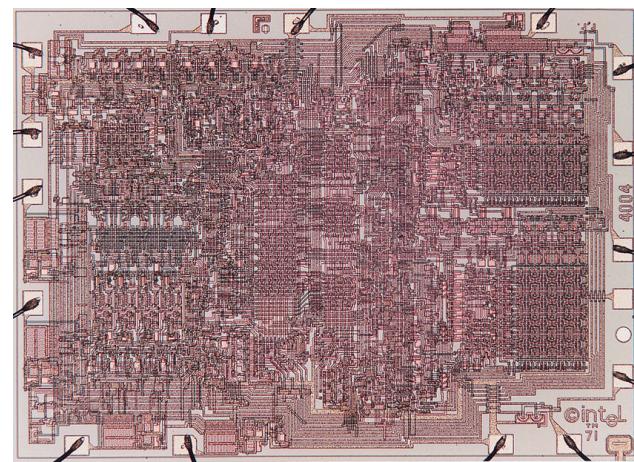
ENIAC

ENIAC (Electronic Numerical Integrator and Computer):

- The computer consisted of 18,000 vacuum tubes and 1500 relays. The ENIAC weighted 30 tons and consumed 140 kilowatts of power. Architecturally the machine had 20 registers, each capable of holding a 10-digit decimal number.
- The ENIAC was programmed using 6000 multiposition switches and connecting a multitude of sockets.

Intel 4004

- Maximum [clock rate](#) is 740 [kHz](#) in 1971
- 46250 to 92500 instructions per second.
- Separate program and data storage.
 - 12-bit addresses
 - 8-bit instructions
 - 4-bit data word
- Can address 640 bytes of RAM, stored as 1280 4-bit "characters"
- Can address 32,768 bits of ROM, arranged as 4096 8-bit bytes
 - [Instruction set](#) had 46 instructions (of which 41 were 8 bits wide and 5 were 16 bits wide)
- 16 registers of 4 bits each
- Had 2300 transistors



Intel Core I7

Clock Rate 1800 - 4900
MHz

Level 1 Cache 256 KB

Level 2 Cache 1 MB

Level 3 Cache 8 MB

Number of Cores 4

8 registers in 32-bit mode
and 16 registers in 64-bit
mode.

64 Bit Word Size

Has ~3 Billion transistors

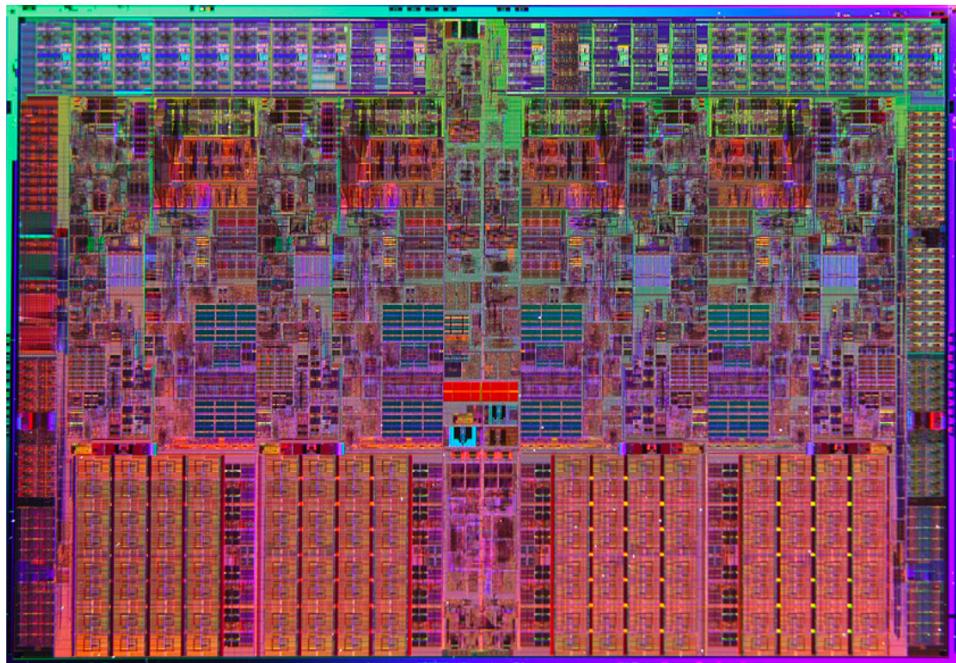


Figure 1.13 Photograph of an Intel Core i7 microprocessor die, which is evaluated in Chapters 2 through 5. The dimensions are 18.9 mm by 13.6 mm (257 mm²) in a 45 nm process. (Courtesy Intel.)

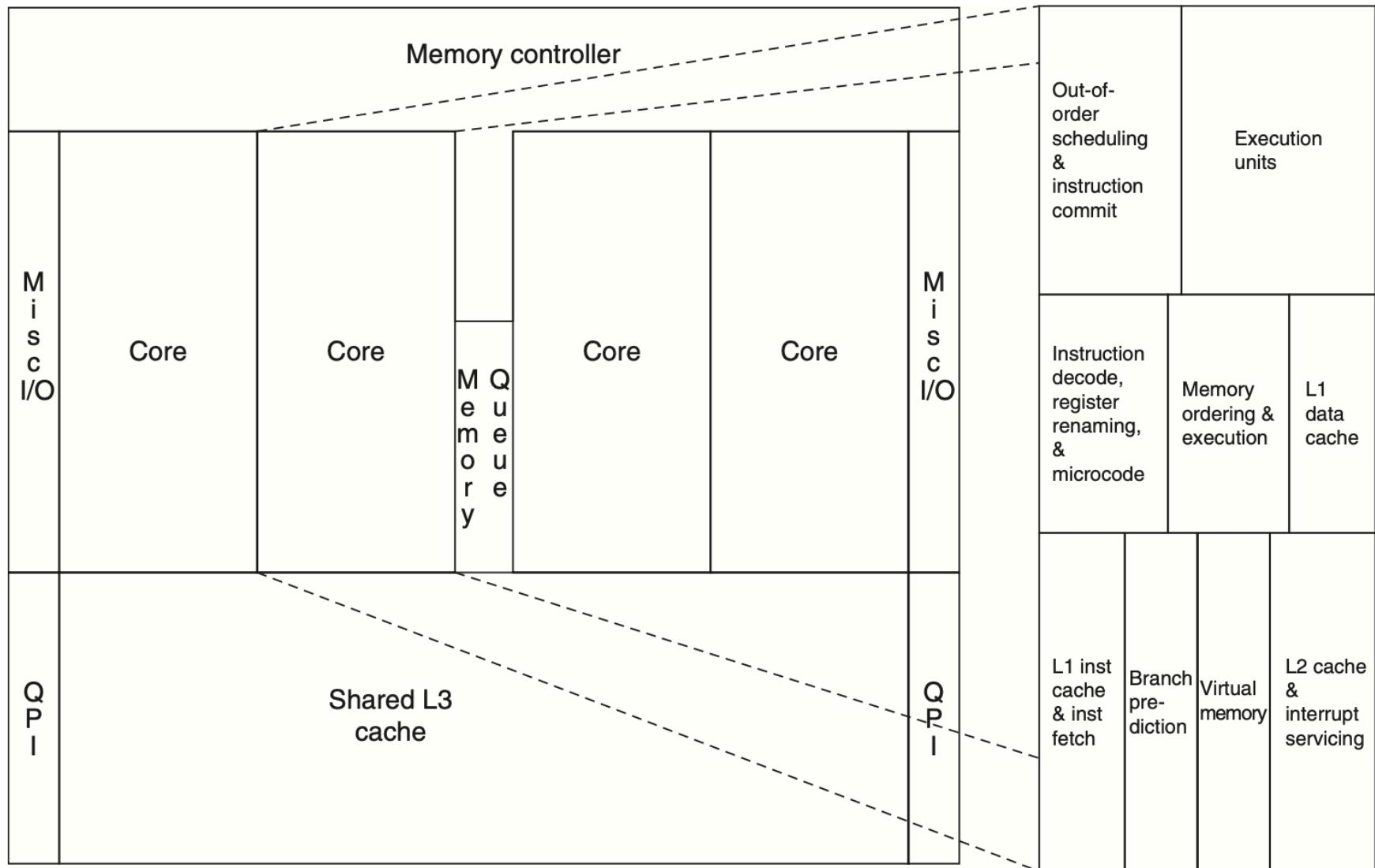


Figure 1.14 Floorplan of Core i7 die in Figure 1.13 on left with close-up of floorplan of second core on right.

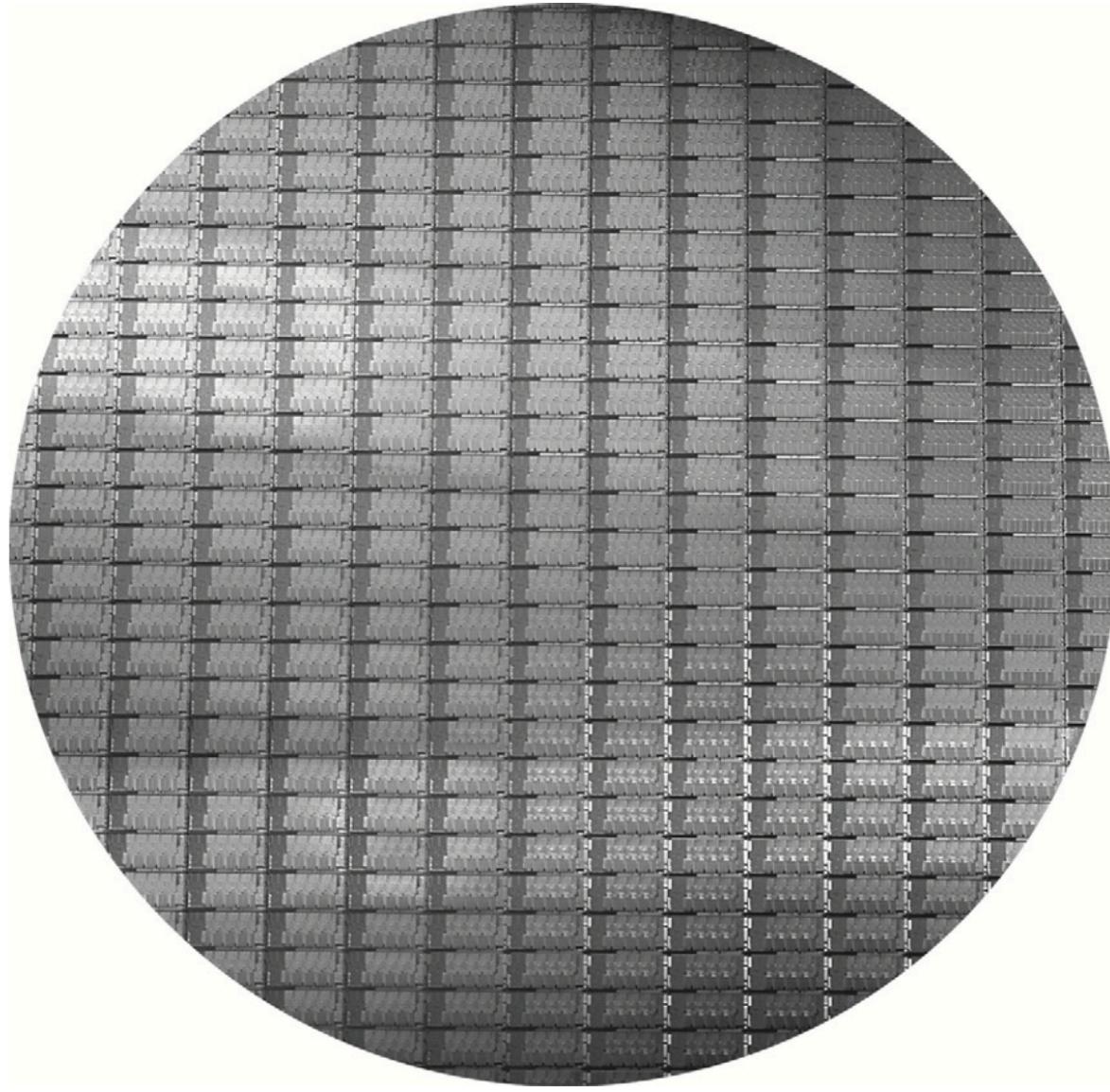


Figure 1.15 This 300 mm wafer contains 280 full Sandy Bridge dies, each 20.7 by 10.5 mm in a 32 nm process. (Sandy Bridge is Intel's successor to Nehalem used in the Core i7.) At 216 mm^2 , the formula for dies per wafer estimates 282. (Courtesy Intel.)

So Architecture is

- Understanding a set of requirements and using your knowledge to produce an Architecture with *abstractions* that can be built in h/w and or s/w using appropriate *technologies*, including *programming languages, systems, (DB, cache, CPUs/ISAs), Operating Systems and Protocols (network and application)*