CLASSES OF PARALLELISM AND PARALLEL ARCHITECTURE.

LLP DLP TLP RLP

parallelism at multiple levels is driving force for computer design.

2 kinds of parallelism in applications.

* data level parallelism
- many data items can be operated upon at the same time.

* task level parallelism
tasks of work are created. that can operate independently and largely in parallel.

computer hardware exploits these 2 parallelism (2 kinds of applications in 4 major ways).

1. instruction level parallelism exploits DLP at modest levels, with compiler help using like pipelining and speculative execution.

2. data level parallelism is exploited by vector processors. and multimedia instruction sets exploit data level parallelism by applying a single instruction to a collection of data in parallel

3. thread level parallelism exploits either data level parallelism of task level parallelism in a tightly coupled hardware module, that allows for interaction between parallel threads.

4. request level parallelism exploits parallelism among largely decoupled tasks, specified by the programmer or operating system.

michael flynn (prof. @ stanford) (1966)
sisd simd misd mimd.
canonical.

flynn (1966) looked at parallelism in instruction and data streams:

1. single instruction stream. } uniprocessor.
single data stream. }

can also exploit instruction level parallelism. and we can also have superscalar systems. and you could also have speculative execution.

2. Single instruction stream
Multiple data stream.

this can exploit data level parallelism.

3. multiple instruction stream, but single data stream.

there is no commercial system of this model.

4. multiple instruction stream.
multiple data stream.

more flexible than simd but inherently more expensive than simd.

tightly coupled mimd architectures exploit (there may be need for interaction, communication), thread level parallelism and loosely coupled mimd architectures are clusters and warehouse scale computers.

architecture = ISA
+ organization of system (micro arch.)
+ hardware realization.

defining computer architecture

the computer designer will determine what attributes are important for a new computer to maximize performance and energy use / efficiency, at the same time cost should not be too high.
while staying within cost, power, and availability constraints.

jan issue of ieee spectrum.

this task has many aspects:-

- instruction set design,
- functional organization,
- logic design
- implementation.

the implementation may encompass ic design, packaging, power and cooling,

ibm 390 systems had cool water cooling.

optimizing design require familiarity with a wide range of technologies
from compilers, and OS to logic design

a few decades ago, computer architecture were referred to as instruction set design,

and. (right) = implementation.

ISA serves as a boundary between the software & hardware.

ARM ~ 14 billion chips sold
~ 50 times x86 processors.

RISC processors are refferred to as load and store architecture

Classes of ISA

Nearly all ISAs are classified as general purpose register architectures. operands are either in registers or memory locations.

8086 had 16 general purpose registers, and they can also hold 16 floating point data.

RISCV has 32 general purpose registers and 32 floating point registers.

register memory ISA.

1955 onwards all ISAs are load-store

2 memory addressing.

these days we have byte addressing.

ARMv8 - aligned. RISCV any byte, x86 } do not require alignment.

3 addressing modes.

(imm. reg.).

4. types & sizes of operands.

8-bit operands ascii
16-bit code unicode (UC2 word).
32-bit integer or word.
64-bit double word. or long integer.
IEEE-754 floating point system.
32 bit single precision
64 bit double precision
80 bit extended double precision.

5. Operations.

data transfer's for operations.
arithmetic
logical.
control
floating point operations.

6. control flow instructions.

conditional branches.
unconditional branches. (jumps) → pc relative
procedure call
return.

classmate
Date 11.1.2020
Page 7.

7. encoding of ISA.

fixed length → implementation is much simpler.
and variable length. → size of code could be compact.

architecture - ISA, organization, hardware.

computer architects design computer to meet functional requirements and in addition must not be too costly, power, performance, availability.

end use / cmp / users become producers

architects also aware of technology development & use of computers

TRENDS IN TECHNOLOGY

ISA survival depends on technological development. as there are technological development, ISA may change. it may be affected by rapid changes in technology.

earlier we had instructions of variable length. since 1985 RISC arch. were introduced, size of instructions remains same.

IIT madras, TATAs 9000 arch., The arch. of that processor remain same for 50 years. The IBM mainframe survived nearly 50 years and the designers must be aware of the technological developments, rapid changes in implementation technology.

there are 5 implementation technologies, and as per the book, they change at a dramatic pace. they are very critical for IC logic technology.

moore's law - dennard scaling (2004) after 30 years.

explore
user more

transistor density has been increasing about 35% /years and die size is less predictable and growth has been slower at 10 - 20% /year.

= semiconductor DRAM
SDRAM 8GB DRAM was shipped in 2014. 16 GB RAM may have been introduced in 2019. 32 GB RAM may not come.

= semiconductor flash (EEPROM)
used in personal mobiled devices.

= magnetic disk technology
3600 rpm → 15000 rpm, size come down to 3.5", could be 2 platters. one company working on heat assisted magnetic recovery. don't have yet commercial technology

= network technology
10 Mbits/s → 10 Gbits/s.
when ethernet introduced.

PERFORMANCE TRENDS

① bandwidth
② throughput
③ ~~performance~~ latency.

= scaling of transistor performance.
prof. anant agarwal "Tilera" company
64 core step processors — 07 nm chips underway.

TRENDS IN POWER AND ENERGY IN ICs.

energy is the biggest challenge, if power is supplied a continuous amount of time and power must be brought from the source.

power must be brought in and distributed around the chips, modern processors have hundreds of pins & mutiple interconnect layers. ones power as heat.

then you look at power and energy, we have we have to worry about performance.

3 primary concerns
= maximum power ever required
= sustained power
= thermal design power (TDP)
which takes care of cooling also

max. power ≈ 1.5 × TDP

* energy and efficiency
many cores not used, some processors many not used all the time - dark silicon comp).

* CMOS

dynamic energy imp. supply voltage come down or current come down, difference between on and off current also has come down.

TRENDS IN COST

fabrication is better understood over a period of time. therefore yield can be improved, impact of time, volume, and commoditization.
= can buy parts and do it.

learning curve. semiconducton are taken with pure silicon rod, sliced - 30 cm dia, all dies may not be good. what causes faulty processes understanding.

volume is a key factor in determining the cost. the cost of an IC is given by the cost of die + ..

cost of IC = cost of die +
cost of testing die +
fabricating +
~~cost of testing after fabrication +~~
cost of packing +
final test

final test yield (~ 90%).

$$\text{cost of die} = \frac{\text{cost of wafer}}{\text{dies per wafer} \times \text{die yield}}$$

$$\text{dies/wafer} = \frac{\pi \times (\text{wafer dia}/2)^2}{\text{die area}}$$

$$- \frac{\pi \times \text{wafer dia}}{\sqrt{2 \times \text{die area}}}$$

$$\text{die yield} = \text{wafer yield} \times \frac{1}{(1 + \text{defects/area} \times \text{die area})^n}$$

for 28nm devices in 2017 = 7.5 - 9.5

16nm = 10 - 14

cost of manufacturing depends upon CAPEX, OPEX dependability

earlier pins were vulnerable, and fault may occur over comm. channels, and this was referred to as conventional wisdom. it changed when devices became very small. there could be transient faults in devices or permanent faults.

and to overcome these faults, we have service level agreements and service level faults.

mean time between failures

= mean time to failure + mean time to repairs

MEASURING, REPORTING, AND SUMMARIZING PERFORM.

performance depends upon the response time. we have the execution time and throughput, and also depends upon workload.

then we have benchmarks. early days kernels used, toy programs, synthetic benchmarks; EDC systems in the 70s. @ kanpur main frame systems. time. sharing system. IIT mumbai, kgp russian systems.

subsequently we have divrysione benchmark suite, also have another EEMBC electronic design new embedded microprocessor benchmark consortium.

programs are in the area of automating industries consumer netcoorking office automation and tele comm.

SPEC system performance evaluation corporation benchmarks 1989 1992 1995 2000 2006 2007.

performance depends upon clock cycle time, and how many cycles per instruction. time it takes depends on no. of instructions.

talk about how many instructions per clock.

QUANTITATIVE PRINCIPLES OF COMPUTER DESIGN

take advantage of parallelism - some scientific applications already - you have to write multicore programs, take advantage of parallelism.

PRINCIPLE OF LOCALITY

= temporal locality
recent instruction may be executed again, like in for loop.

= spatial locality
next instruction may soon be executed. many applications will have 10-15x.