



5118007-02 Computer Architecture

Introduction & Backgrounds

(Ch. 1 Computer Abstractions and Technology)

5 Mar 2024

Shin Hong

Learning Objectives

- get an overview of the general structure of computer
- understand key ideas in computer system designs
 - abstraction (instruction set architecture)
 - memory hierarchy
- comprehend different notions of computer performance

George Boole



Formulate a calculus of reasoning

- Claim that logic should be considered as a branch of math, rather than a part of philosophy
- Argue that there are math laws to express the operation of human mind
- Showed that Aristotle's syllogistic logic could be rendered as algebraic equitation

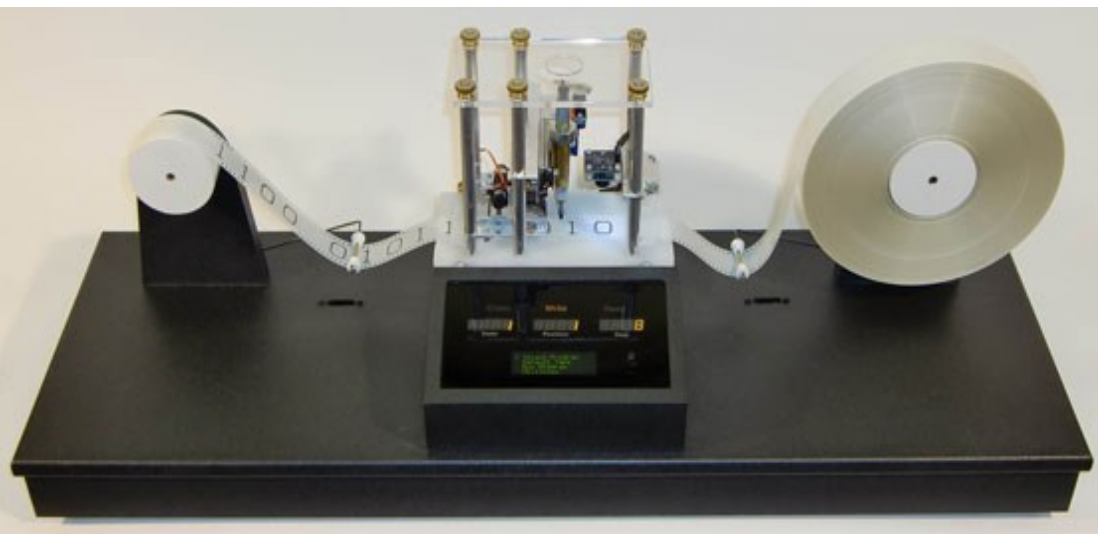
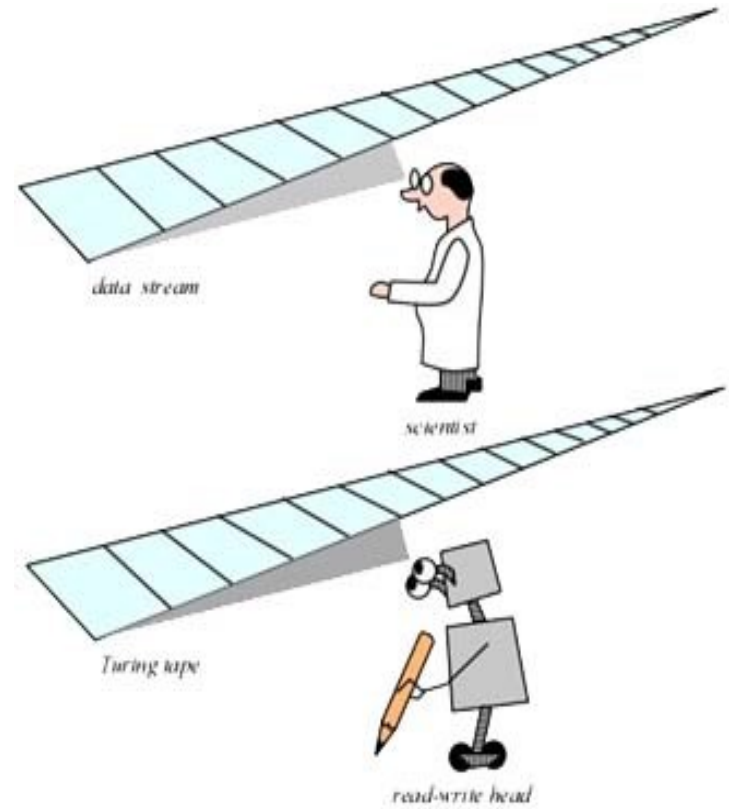
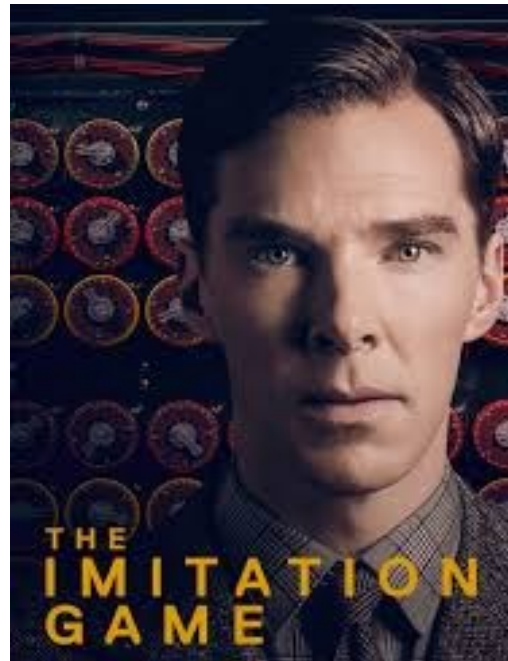
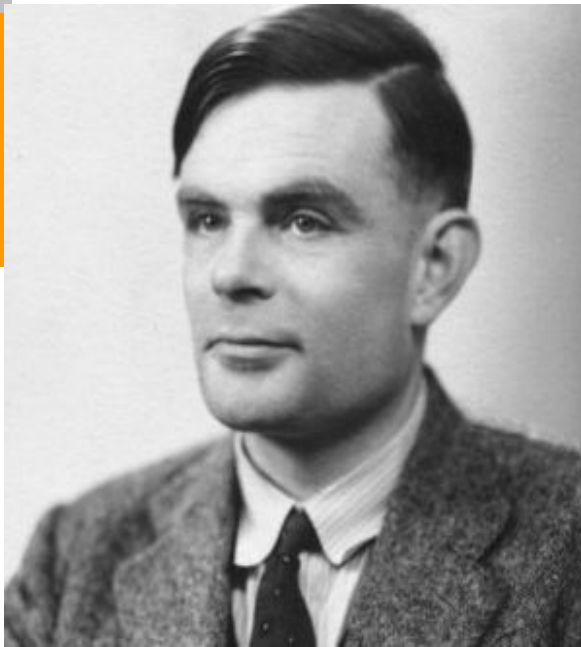
George Boole (1815--1864)

A Brief History of Computing by G. O'Regan

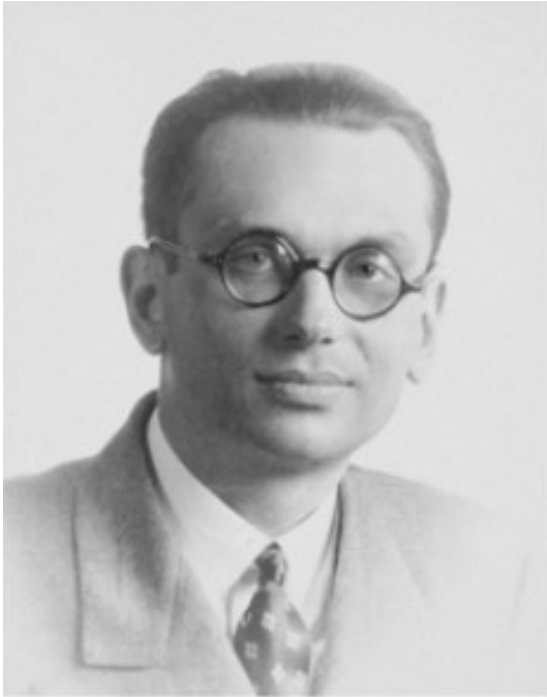
Charles Babbage

Diagram for the computation by the Engine of the Members of Bessel's. See Note G. (page 44 of 49)

Number of Operations.	Nature of Operations.	Transfer and other operations.	Transfer and other operations.	Indication of change in the value of the result.	Statement of Results.	Name.										Working Variables.		
						x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
1	$x_1 = 0.724441860$				$x_1 = 0.724441860$													
2	$x_2 = 0.724441860$				$x_2 = 0.724441860$													
3	$x_3 = 0.724441860$				$x_3 = 0.724441860$													
4	$x_4 = 0.724441860$				$x_4 = 0.724441860$													
5	$x_5 = 0.724441860$				$x_5 = 0.724441860$													
6	$x_6 = 0.724441860$				$x_6 = 0.724441860$													
7	$x_7 = 0.724441860$				$x_7 = 0.724441860$													
8	$x_8 = 0.724441860$				$x_8 = 0.724441860$													
9	$x_9 = 0.724441860$				$x_9 = 0.724441860$													
10	$x_{10} = 0.724441860$				$x_{10} = 0.724441860$													
11	$x_{11} = 0.724441860$				$x_{11} = 0.724441860$													
12	$x_{12} = 0.724441860$				$x_{12} = 0.724441860$													
13	$x_{13} = 0.724441860$				$x_{13} = 0.724441860$													
14	$x_{14} = 0.724441860$				$x_{14} = 0.724441860$													
15	$x_{15} = 0.724441860$				$x_{15} = 0.724441860$													
16	$x_{16} = 0.724441860$				$x_{16} = 0.724441860$													
17	$x_{17} = 0.724441860$				$x_{17} = 0.724441860$													
18	$x_{18} = 0.724441860$				$x_{18} = 0.724441860$													
19	$x_{19} = 0.724441860$				$x_{19} = 0.724441860$													
20	$x_{20} = 0.724441860$				$x_{20} = 0.724441860$													
21	$x_{21} = 0.724441860$				$x_{21} = 0.724441860$													
22	$x_{22} = 0.724441860$				$x_{22} = 0.724441860$													
23	$x_{23} = 0.724441860$				$x_{23} = 0.724441860$													
24	$x_{24} = 0.724441860$				$x_{24} = 0.724441860$													
25	$x_{25} = 0.724441860$				$x_{25} = 0.724441860$													
26	$x_{26} = 0.724441860$				$x_{26} = 0.724441860$													
27	$x_{27} = 0.724441860$				$x_{27} = 0.724441860$													
28	$x_{28} = 0.724441860$				$x_{28} = 0.724441860$													
29	$x_{29} = 0.724441860$				$x_{29} = 0.724441860$													
30	$x_{30} = 0.724441860$				$x_{30} = 0.724441860$													
31	$x_{31} = 0.724441860$				$x_{31} = 0.724441860$													
32	$x_{32} = 0.724441860$				$x_{32} = 0.724441860$													
33	$x_{33} = 0.724441860$				$x_{33} = 0.724441860$													
34	$x_{34} = 0.724441860$				$x_{34} = 0.724441860$													
35	$x_{35} = 0.724441860$				$x_{35} = 0.724441860$													
36	$x_{36} = 0.724441860$				$x_{36} = 0.724441860$													
37	$x_{37} = 0.724441860$				$x_{37} = 0.724441860$													
38	$x_{38} = 0.724441860$				$x_{38} = 0.724441860$													
39	$x_{39} = 0.724441860$				$x_{39} = 0.724441860$													
40	$x_{40} = 0.724441860$				$x_{40} = 0.724441860$													
41	$x_{41} = 0.724441860$				$x_{41} = 0.724441860$													
42	$x_{42} = 0.724441860$				$x_{42} = 0.724441860$													
43	$x_{43} = 0.724441860$				$x_{43} = 0.724441860$													
44	$x_{44} = 0.724441860$				$x_{44} = 0.724441860$													
45	$x_{45} = 0.724441860$				$x_{45} = 0.724441860$													
46	$x_{46} = 0.724441860$				$x_{46} = 0.724441860$													
47	$x_{47} = 0.724441860$				$x_{47} = 0.724441860$													
48	$x_{48} = 0.724441860$				$x_{48} = 0.724441860$													
49	$x_{49} = 0.724441860$				$x_{49} = 0.724441860$													
50	$x_{50} = 0.724441860$				$x_{50} = 0.724441860$													
51	$x_{51} = 0.724441860$				$x_{51} = 0.724441860$													
52	$x_{52} = 0.724441860$				$x_{52} = 0.724441860$													
53	$x_{53} = 0.724441860$				$x_{53} = 0.724441860$													
54	$x_{54} = 0.724441860$				$x_{54} = 0.724441860$													
55	$x_{55} = 0.724441860$				$x_{55} = 0.724441860$													
56	$x_{56} = 0.724441860$				$x_{56} = 0.724441860$													
57	$x_{57} = 0.724441860$				$x_{57} = 0.724441860$													
58	$x_{58} = 0.724441860$				$x_{58} = 0.724441860$													
59	$x_{59} = 0.724441860$				$x_{59} = 0.724441860$													
60	$x_{60} = 0.724441860$				$x_{60} = 0.724441860$													
61	$x_{61} = 0.724441860$				$x_{61} = 0.724441860$													
62	$x_{62} = 0.724441860$				$x_{62} = 0.724441860$													
63	$x_{63} = 0.724441860$				$x_{63} = 0.724441860$													
64	$x_{64} = 0.724441860$				$x_{64} = 0.724441860$													
65	$x_{65} = 0.724441860$				$x_{65} = 0.724441860$													
66	$x_{66} = 0.724441860$				$x_{66} = 0.724441860$													
67	$x_{67} = 0.724441860$				$x_{67} = 0.724441860$													
68	$x_{68} = 0.724441860$				$x_{68} = 0.724441860$													
69	$x_{69} = 0.724441860$				$x_{69} = 0.724441860$													
70	$x_{70} = 0.724441860$				$x_{70} = 0.724441860$													
71	$x_{71} = 0.724441860$				$x_{71} = 0.724441860$													
72	$x_{72} = 0.724441860$				$x_{72} = 0.724441860$													
73	$x_{73} = 0.724441860$				$x_{73} = 0.724441860$													
74	$x_{74} = 0.724441860$				$x_{74} = 0.724441860$													
75	$x_{75} = 0.724441860$				$x_{75} = 0.724441860$													
76	$x_{76} = 0.724441860$				$x_{76} = 0.724441860$													
77	$x_{77} = 0.724441860$				$x_{77} = 0.724441860$													
78	$x_{78} = 0.724441860$				$x_{78} = 0.724441860$													
79	$x_{79} = 0.724441860$				$x_{79} = 0.724441860$													
80	$x_{80} = 0.724441860$				$x_{80} = 0.724441860$													
81	$x_{81} = 0.724441860$				$x_{81} = 0.724441860$													
82	$x_{82} = 0.724441860$				$x_{82} = 0.724441860$													
83	$x_{83} = 0.724441860$				$x_{83} = 0.724441860$													
84	$x_{84} = 0.724441860$				$x_{84} = 0.724441860$													
85	$x_{85} = 0.724441860$				$x_{85} = 0.724441860$													
86	$x_{86} = 0.724441860$				$x_{86} = 0.724441860$													
87	$x_{87} = 0.724441860$				$x_{87} = 0.724441860$													
88	$x_{88} = 0.724441860$				$x_{88} = 0.724441860$													
89	$x_{89} = 0.724441860$				$x_{89} = 0.724441860$													
90	$x_{90} = 0.724441860$				$x_{90} = 0.724441860$													



Mathematical Model of Computing



Kurt Gödel



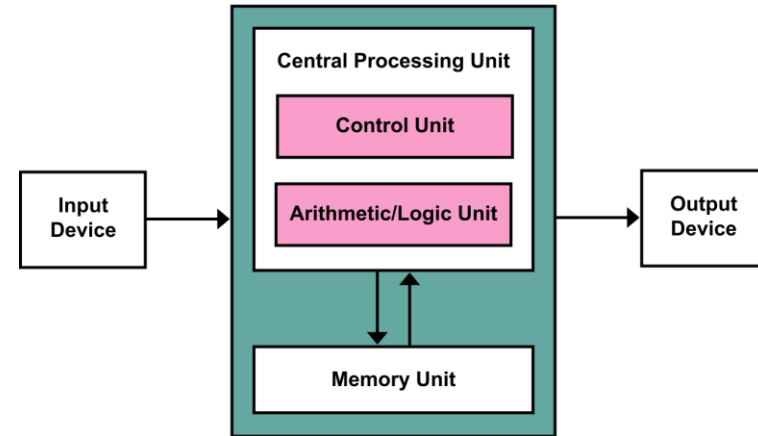
Alan Turing



Alonzo Church

Modern Computer Model

- Memory is a map from addresses to values
- A value is either a number or instruction
 - number
 - instruction
 - receive an input
 - produce an output
 - evaluate an expression over memory addresses
 - assign a value to a memory address
 - jump to a memory address
 - finish
- A processor loads and executes instructions from address



John von Neumann (1903— 1957)

Innovations in Semiconductor Technology

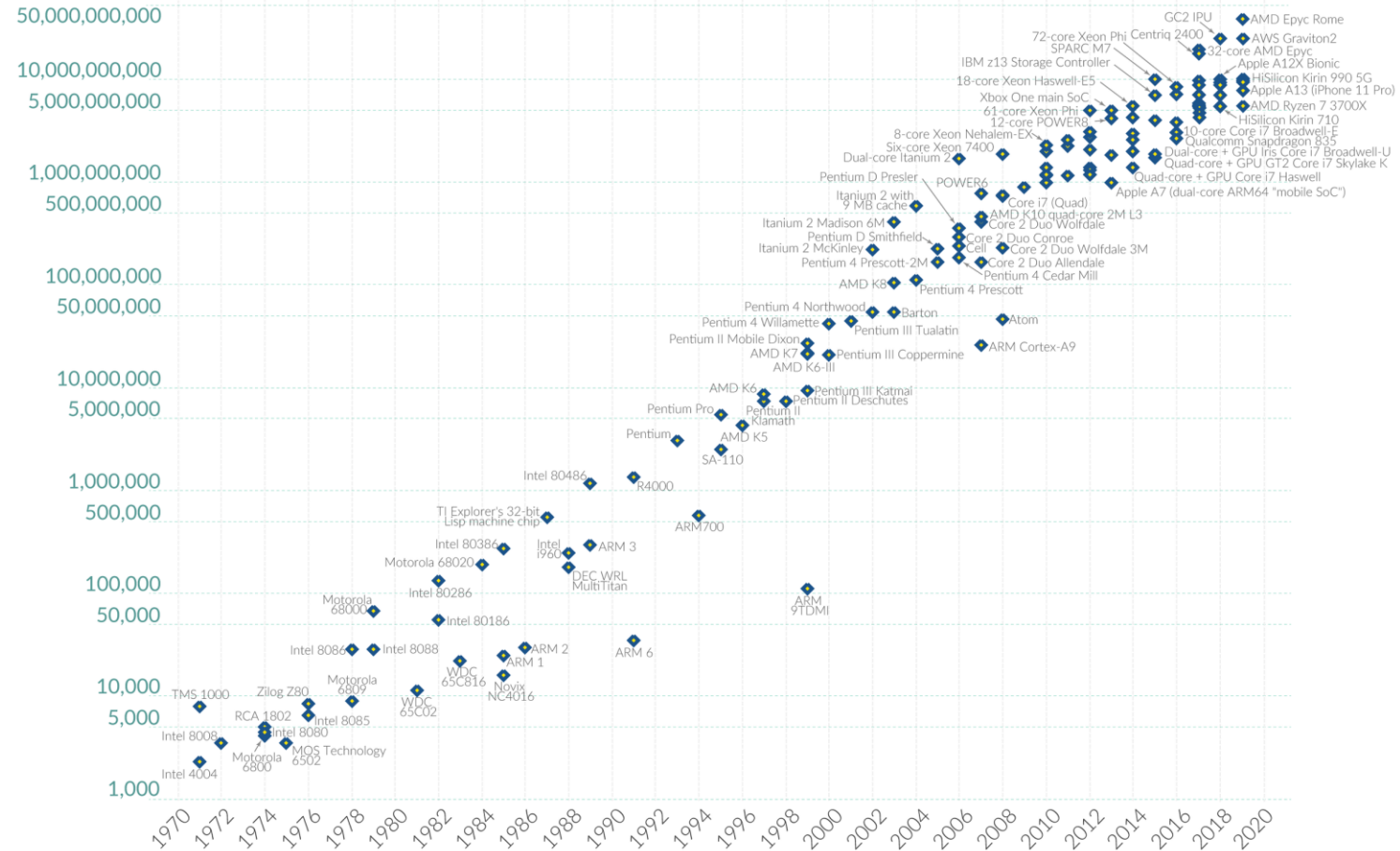
Moore's Law: The number of transistors on microchips doubles every two years

Our World
in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.

This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Transistor count

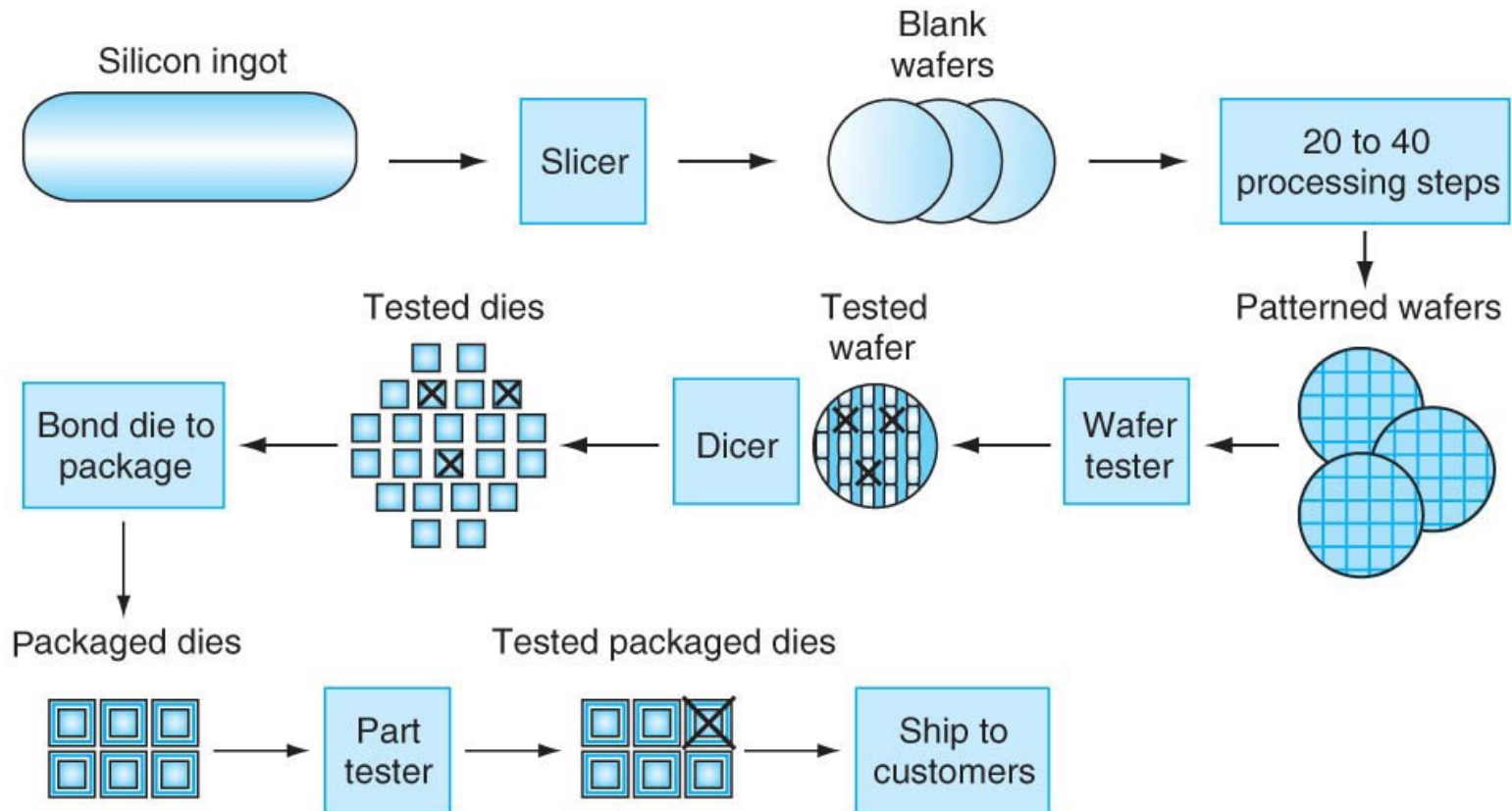


Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

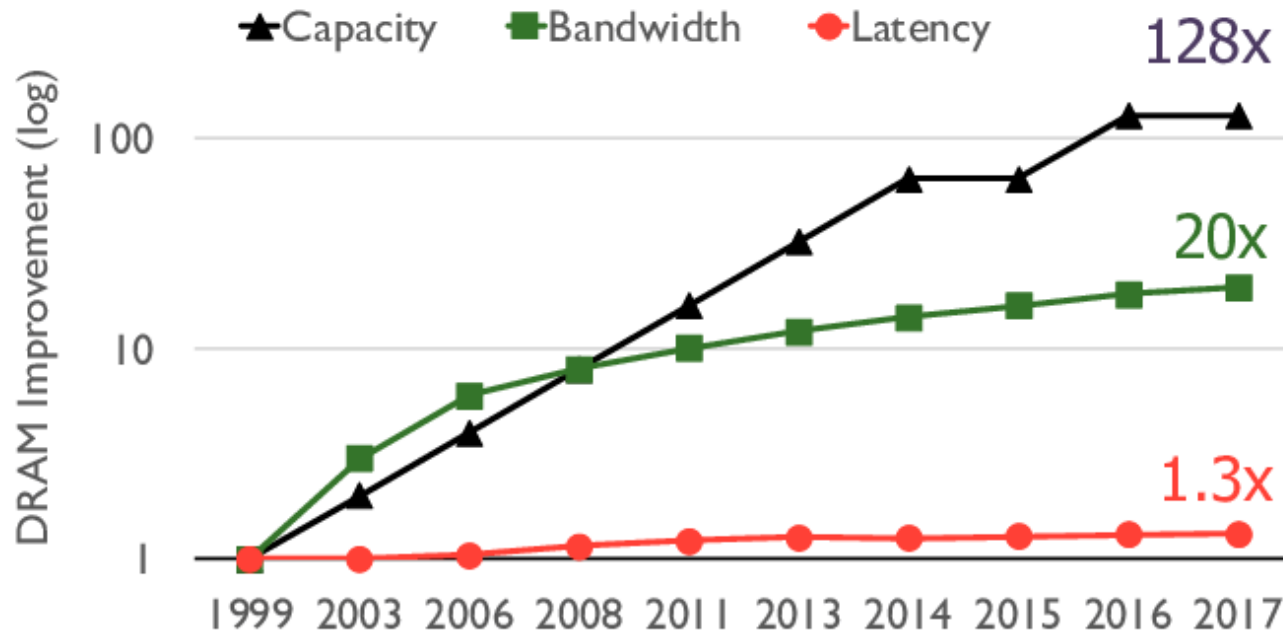
Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Semiconductor Manufacturing Process



Innovations in Semiconductor Technology

DRAM Capacity, Bandwidth & Latency



Innovations in Semiconductor Technology

경제 산업·재계

삼성 2나노 반도체 2025년 양산... 공정 로드맵 첫 공개

'삼성 파운드리 포럼'서 2nm 양산 로드맵 공개

기자 욱기원

수정 2023-06-28 17:29 등록 2023-06-28 17:29



삼성전자가 27일(현지시간) 미국 실리콘밸리에서 개최한 '삼성 파운드리 포럼 2023'에서 최시영 파운드리 사업부 사장이

OpenAI chief looking to raise trillions to reshape se sector: WSJ

f

X

✉

AFP News

9 February 2024 · 2-min read

In this article:

Sam Altman

American venture-c...

Masayoshi Son

Japanese businessm...

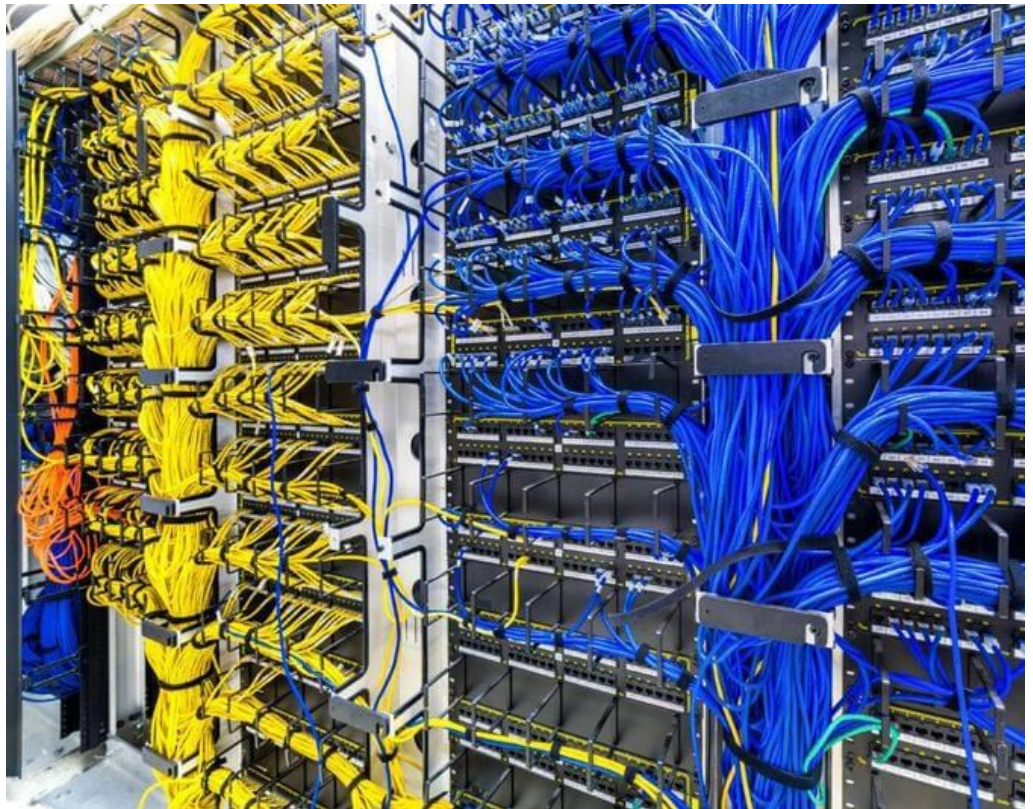
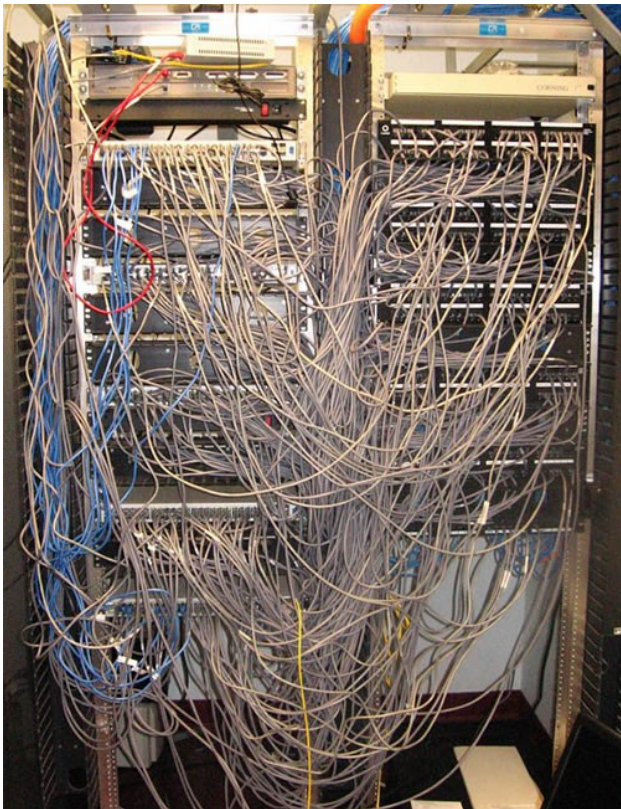


Sam Altman has reportedly held talks with potential investors including the UAE government (Fabrice COFFRINI)

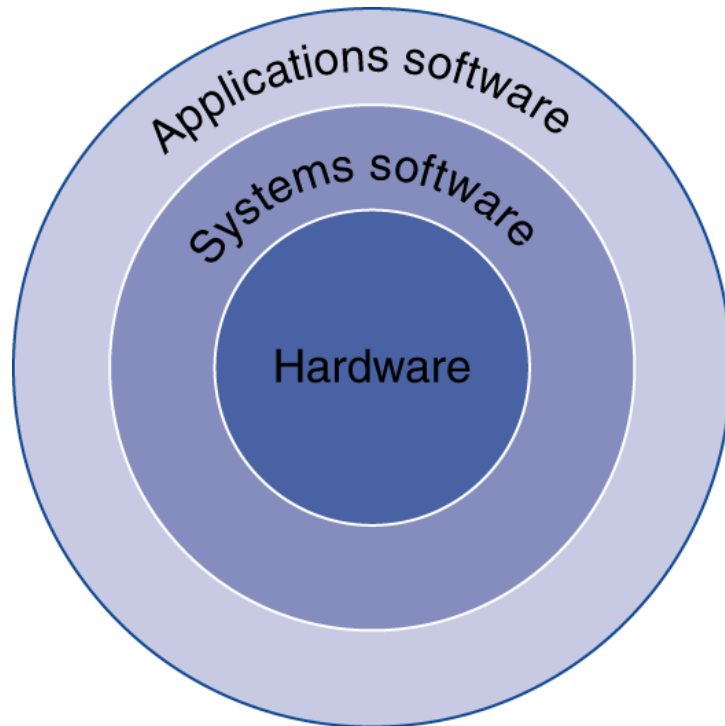
OpenAI chief Sam Altman is seeking to raise trillions

Abstraction

- Define an interface that hides implementation details and offers functional primitives

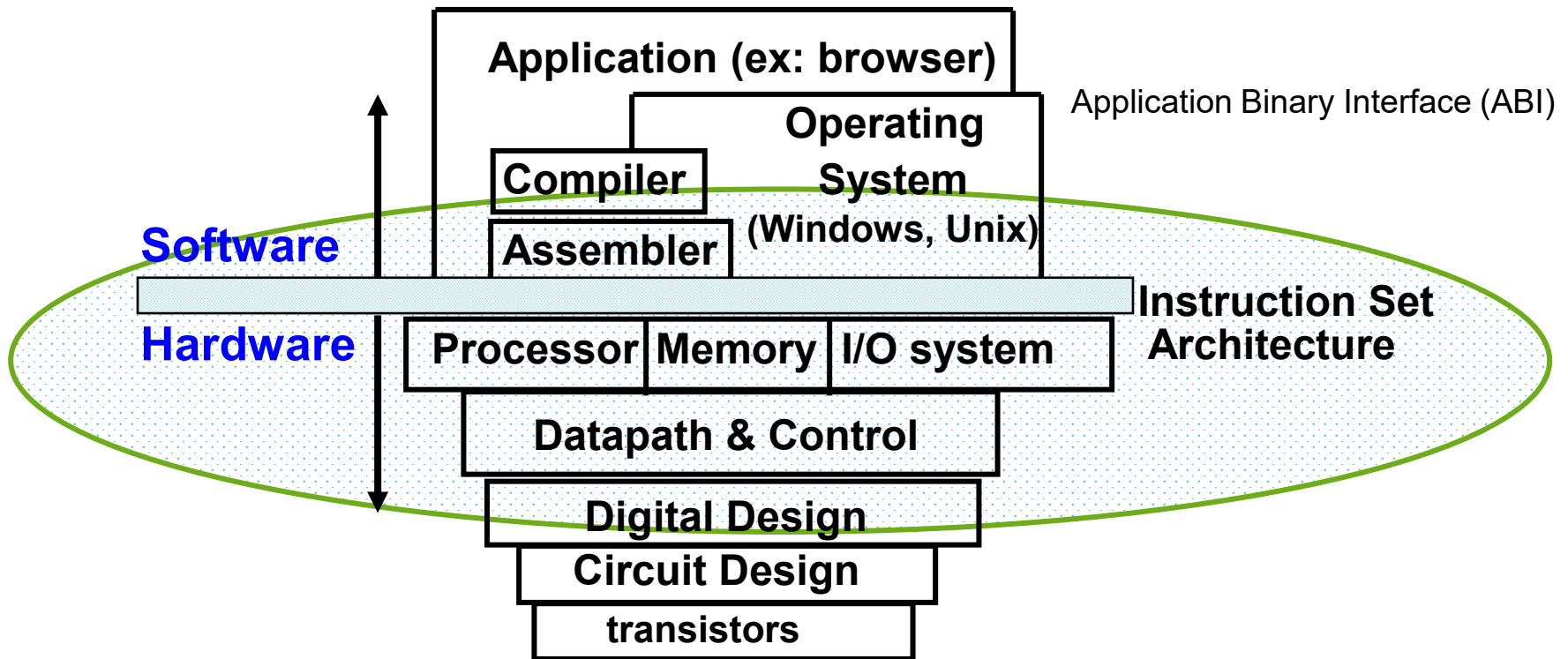


Below Your Program



- Application software
- System software
 - Compiler: translates HLL code to machine code
 - Operating System: service code
 - Handling input/output
 - Managing memory and storage
 - Scheduling tasks & sharing resources
- Hardware
 - Processor, memory, I/O controllers

Cross Section View



Program Code Representations

High Level Language
Program

Compiler

Assembly Language
Program

Assembler

Machine Language
Program

Machine Interpretation

Control Signal
Specification

```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;
```

“instruction”

lw	\$15,	0(\$2)
lw	\$16,	4(\$2)
sw	\$16,	0(\$2)
sw	\$15,	4(\$2)

```
0000 1001 1100 0110 1010 1111 0101 1000  
0000 1001 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
1100 0110 0000 1001 1100 0110 1010 1111
```

High and low signals on control lines

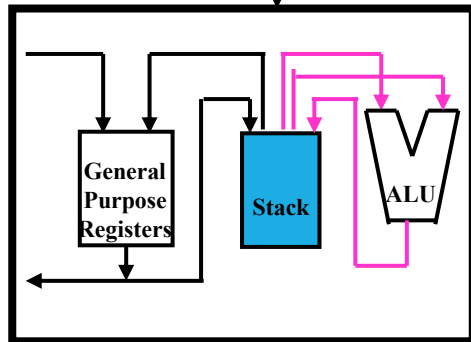
Instruction Set Architecture (ISA)

$C = A + B;$

ISA 1

push A
push B
add
pop C

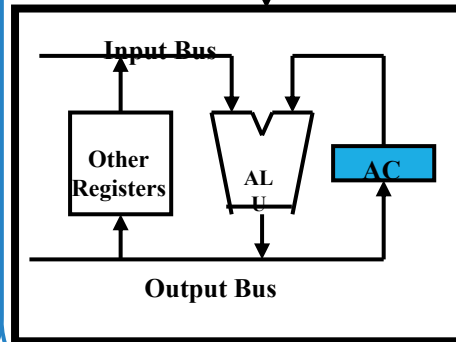
```
1100 1001 1100 0110
1010 1011 0101 1010
1100 0110 1010 1111
0101 1010 0000 1001
```



ISA 2

load A
add B
store C

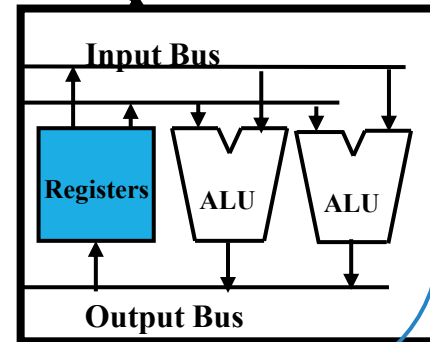
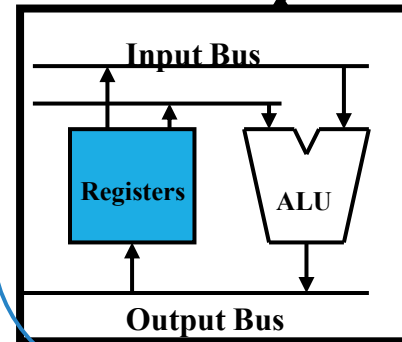
```
0110 1001 1010 0110
1010 1111 0101 1000
1010 0110 1010 1111
```



ISA 3

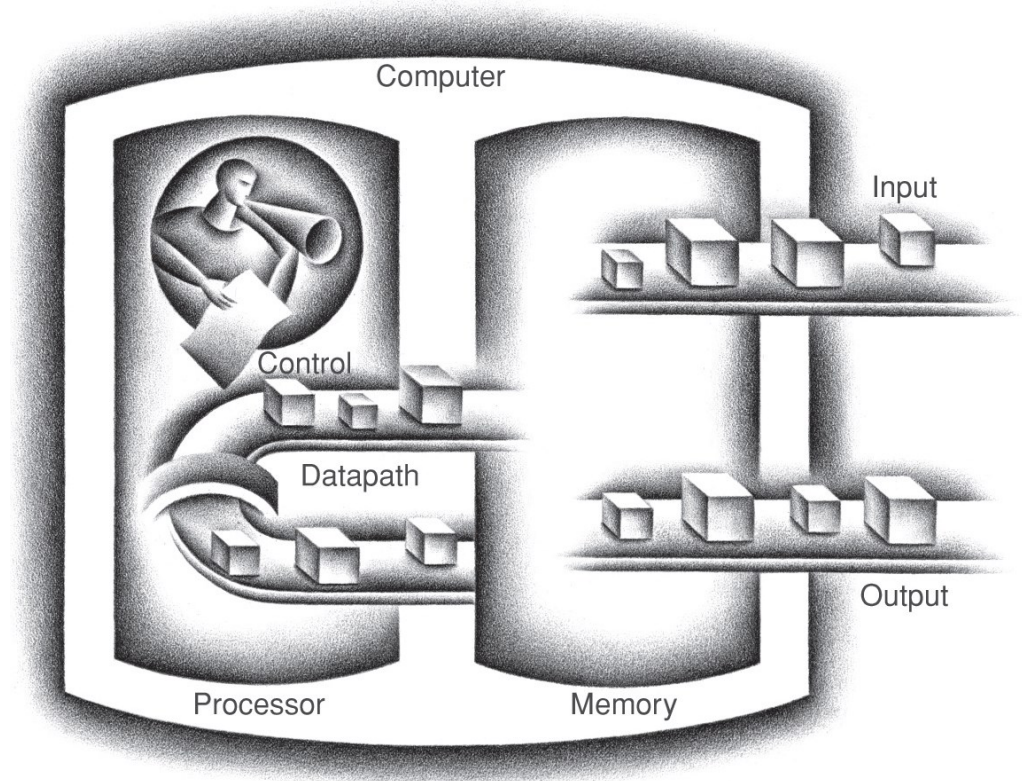
ld R1, A
ld R2, B
add R3, R1, R2
st C, R3

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```



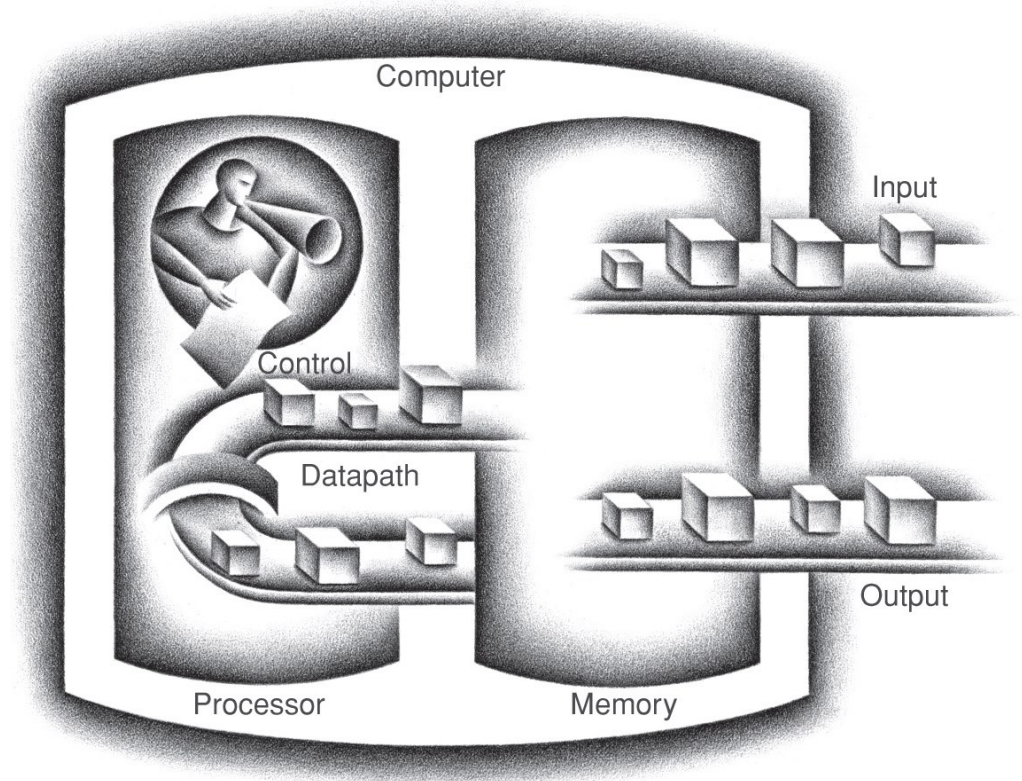
Computer System Organization

- Five conceptual components
 - Input
 - Output
 - Datapath
 - Control
 - Memory



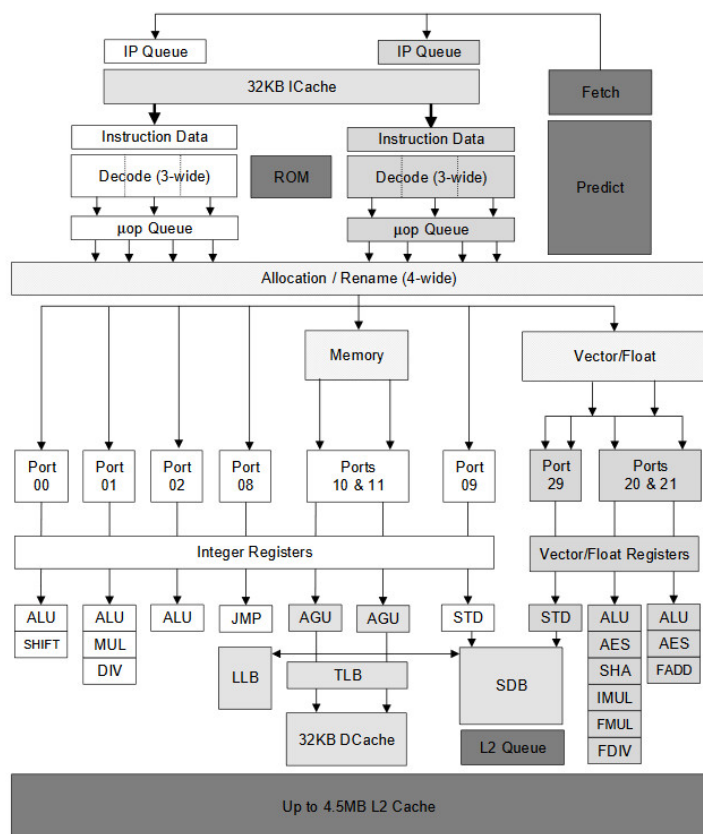
Computer System Organization

- Five conceptual components
 - Input
 - Output
 - Datapath
 - Control
 - Memory

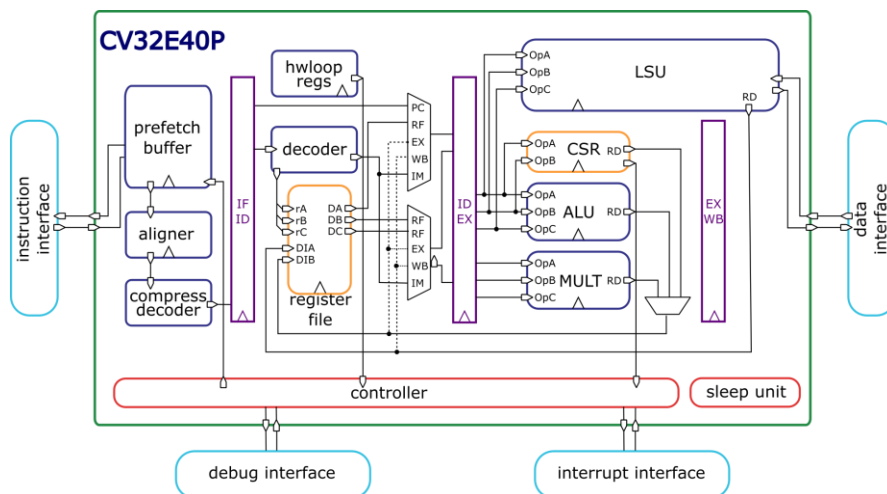


Modern Computer Architecture

- x86/x64



- CV32E40P: 32-bit RISC-V



https://docs.openhwgroup.org/projects/cv32e40p-user-manual/en/cv32e40p_v1.0.0_doc/intro.html