Computation Structures 2014 Yue Zhang

S7 ☐ Turing machines & computability
☐ The von Neumann architectur
☐ A rough sketch of CPU
☐ Three types of beta instructions

• Turing machines & computability

- mathematical functions

  + − * / log cos factorial ...

  Map $X \mapsto Y$

- integer functions $X, Y = \mathbb{Z}$

- can all integer functions be computed? no.

Theory

combinational logic → divisible by 3, sum list integer.
lock 100
infinite input.

^

finite state machines
(take steps) → example: read in string
output ( ) match.
need infinite memory,
not finite state machine

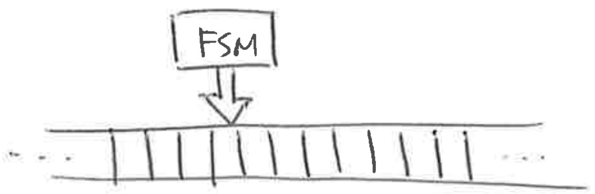
^

Turing machines.

^

All integer functions.

- Turing machines.



infinite tape

i: data on tape

o: write; move.

example: write 1 move left if there is (
move ~~write~~ right reverse bit
if there is) S= L/R/0.

Turing machines are more powerful because of infinite tape.

- It has been shown that Turing Machines have the Same Computation power as recursion <Kleene> and lambda <Church> calculus, another two famous computation tools 50s.

- It has been hypothesized that anything computable by a machine is computable by Turing machine.
  (Church-Turing thesis) Enumerate : what could? real / int ...

- There are many interesting Turing Machines : universal turing machine, the halting machine .

- It has been proved that there are integer functions that Turing machines cannot compute.

(optional proof)

	I_1	I_2	...
FSM1	O_1	O_2	...
FSM2	O_3	Loop	...
⋮	⋮		

Some inputs make Turing machines loop infinitely

There are an infinite number of rows / columns

$T_k(j) \rightarrow FSM_k$ on I_j ; halt(k, j) is computable. from universal TM.

Can build a Turing machine $T_N(x)$ that loops if $T_x(x)$ halts
halts if $T_x(x)$ loops

T_N cannot exist, because $T_N(N)$ is dilema.

Practice —— no infinite tapes
therefore,
Turing machines with finite ~~can~~ must be FSM

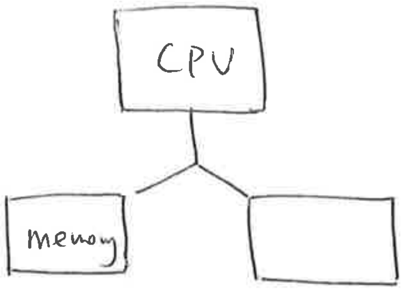
All Computers are FSMs.

- So is it only useful theoretically to study Turing machines?
not really, because of the concept being extended

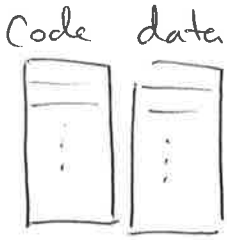
- Universal Turing machines.
  $\forall k, j \quad U(k, j) = T_k(j)$ is computable.
  Useful!
  One machine, simulates all machines.
  k — algorithm (code)
  j — data
  Prototype of computers.

Von Neumann machine

- The current computers are based on Von Neumann machines.
- It models computers in a more engineering way.
- machine



- memory



word seq
8/16/32/64

- CPU repeatedly load code and execte, reading and writing data
- Devices interact with CPU via memory for data interaction,
  and via OS for command interaction.

- Is that all?

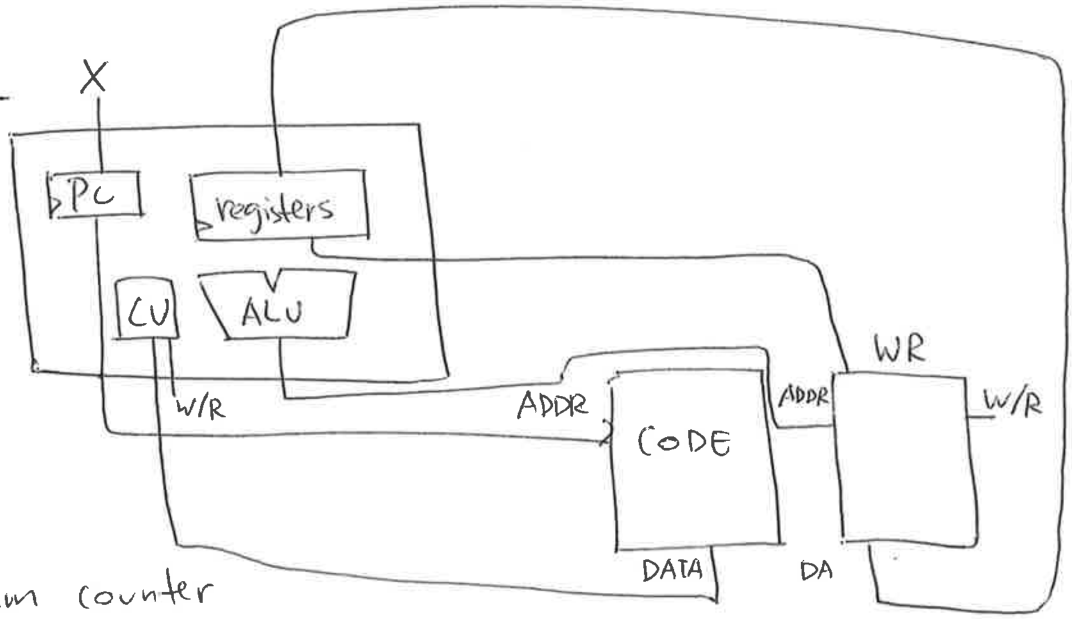
Yes

Showing to monitor is writing to monitor
Controlling robots is writing to robots
reading from the Internet is from URL

all is data/IO

pedagogy: coarse to fine, repeat.

- CPU



- program counter

  points to the code section of the memory
  loading the current instruction
  automatically increment by 1

- control unit

  interprets instructions and decides output signals

- what instructions are?

  B a computer word just like other data
  encodes commands for the CU
  We study the MIT beta system, word = four bytes

- memory: byte ad.

• Three types of beta instructions

ALU

ADD. SUB. MUL. DIV | AND. OR. XOR | SHL. SHR. SHA | CMPEQ. CMPLE. CMPLT
(RA, RB → RC)

~~SHL, SHR, S~~
ADDC. SUBC. MULC, DIVC | ANDC. ORC. XORC | SHLC. SHRC. SHAC | CMPEQC. CMPLEC. CMPLTC
(RA, C → RC)

<change registers ∈ state>

Memory

LD. ST
MEM[RA. OFFSET] ↔ RC

LDR
MEM[PC. OFFSET] → RC

< change memory/registers ∈ state)

PC

BEQ. BNE
RA=0?   PC ← PC+4+OFFSET   RC ← PC+4

JMP
PC ← PC+RA   RC ← PC+4

<change PC. ∈ states).

- Complex state machine
- state { PC, registers, memory }
- devices can have state also.