Lecture 1 August 6th

Note Title 06-08-2011

Turing Machine:

↳ Basic → Single Tape

Complex

↳ Multiple Tapes, Multiple Tape Heads

Could not find a problem that was not solvable by a Turing machine.

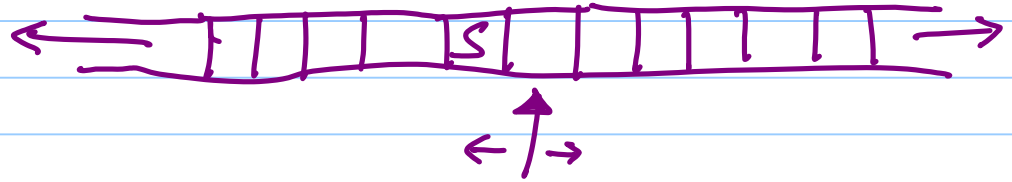
Alonzo Church → Functional Programming

Church-Turing Thesis: Anything that is computable can be computed by a Turing machine

Cannot be computed by a Turing Machine

↳ Can I write a program to find an infinite loop in another program!

Turing Machine

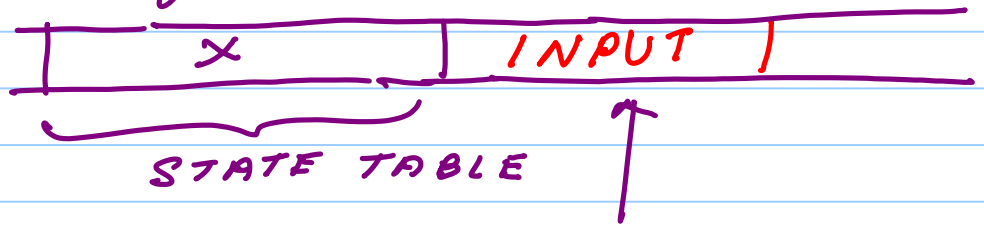


State Table

$S, q \rightarrow S', q', (L|R)$

Program

Universal Turing Machine



State Table Generic.

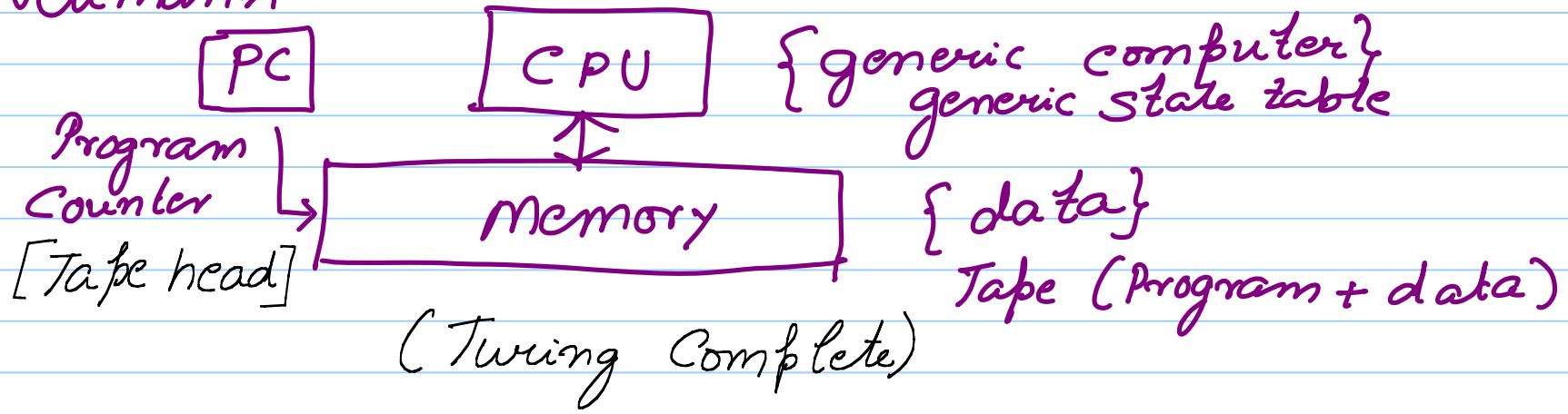
What do we learn?

(i) Program can be stored

(ii) Data can be stored

(iii) We can have a generic computer

Von Neumann



(Turing Complete)

Program → Sequence of instructions.

PC → Program Counter → Points to current Instruction

One instruction Set computer

Most Arithmetic Operations → Subtract is sufficient

sub: $a-b$
add: $a-(0-b)$
mul: $a \times b$
div: sub is not enough
need a way to branch.

```
   6
   79
   87
  ----
  553
 632
 ----
 6873
```

```
23 ) 472 ( 20
     46
     ---
      12
       0
     ---
      12
```

universal instruction : subtract & branch if less than zero
(SUBLE)

SUBLE is Turing Complete. You don't need anything else.

C program: SUB & if & goto

Data processing Instructions:

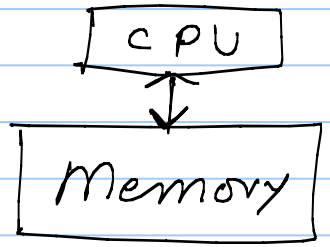
Arithmetic : Add, Subtract, Multiply
Divide

Logical: Bitwise OR, XOR, AND, aAND
NOR.

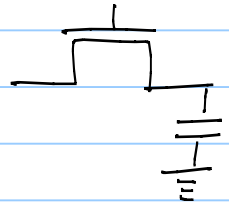
Control Flow Instruction:

Branch (if + goto), call, return

Von Neumann Architecture



DRAM



DRAM memory is extremely slow.
Latencies for a DRAM are very high
100 ns per access.

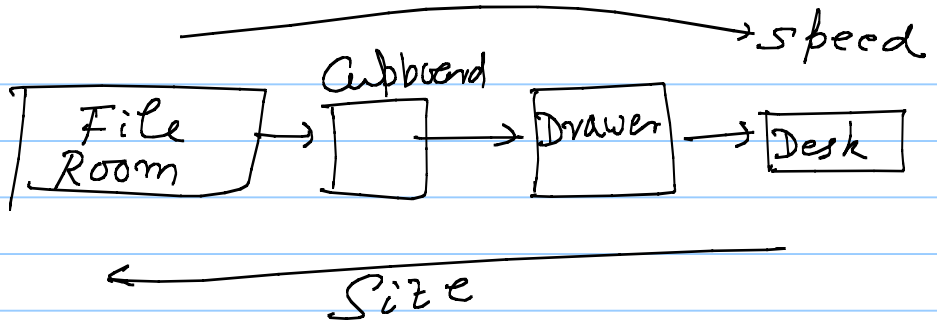
② a = b + c ①
④ d = a + 10 ③
⑥ f = 2 × d ⑤

600 ns

1 clock period = 1/3 ns.

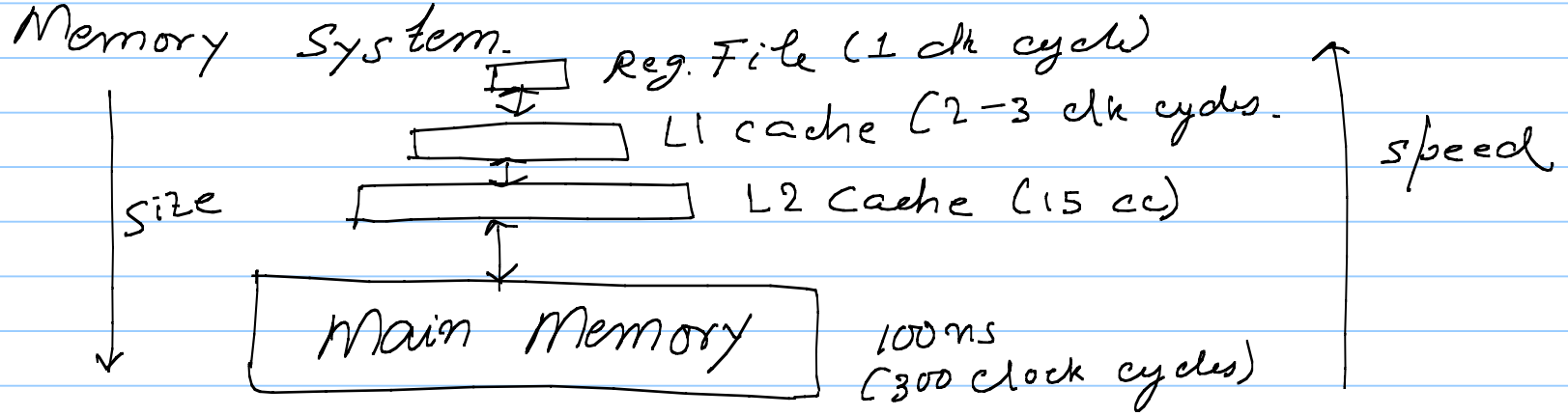
Faster Option ???

Government Office file access protocol.



Temporal Locality.

Memory System.



95 % → RF
4 % → L1
0.9 % → L2
0.1 % → Main Mem.

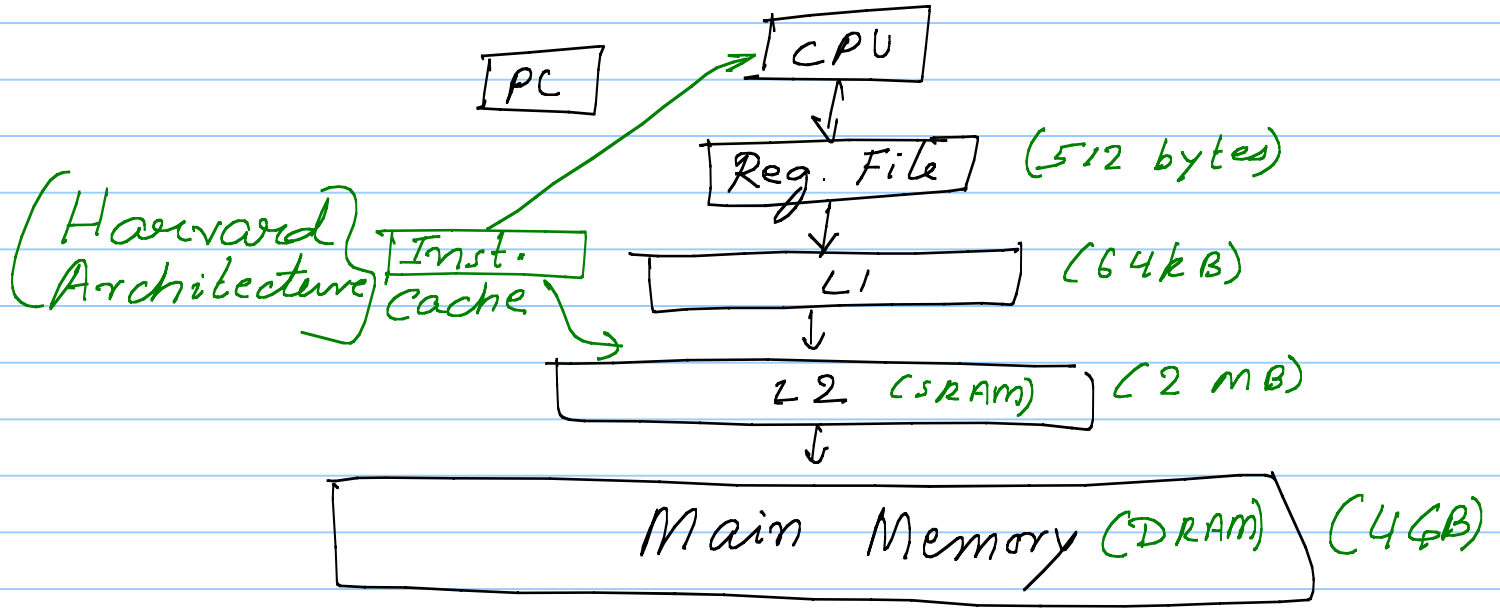
Avg. Latency = $0.95 \times 1 + 0.04 \times 2$
$+ 0.009 \times 15$
$+ 0.001 \times 300$

$= 0.95 + 0.08 + 0.135 + 0.3$

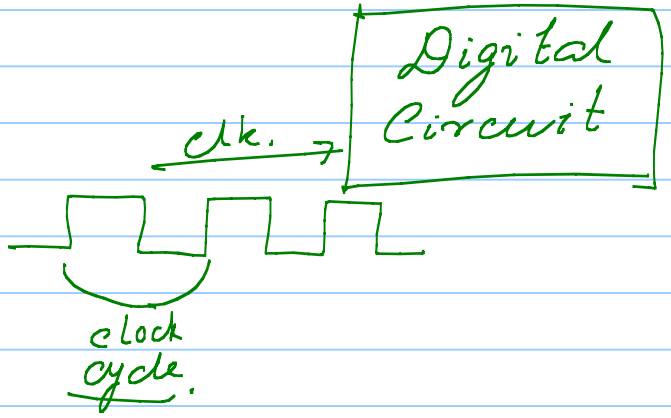
$= 1.465$ clk cycles.

(300 ⟶ 1.5
200× speedup)

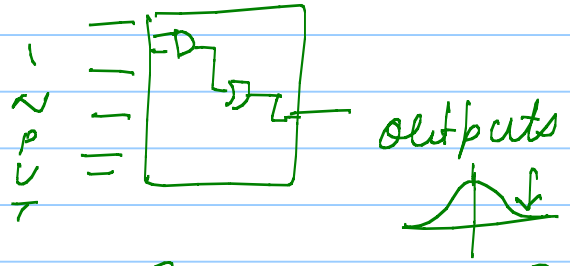
Von Neumann with hierarchical Memory



Clock Cycle:

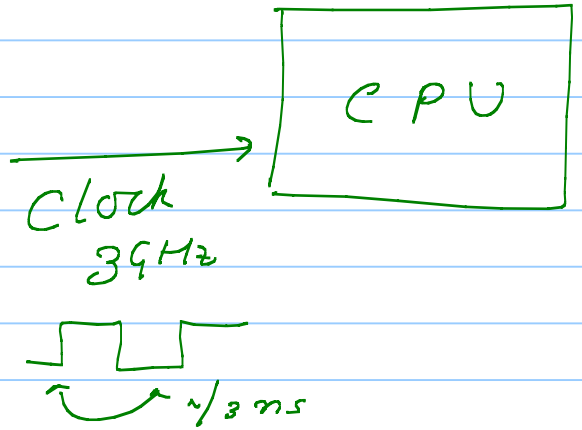


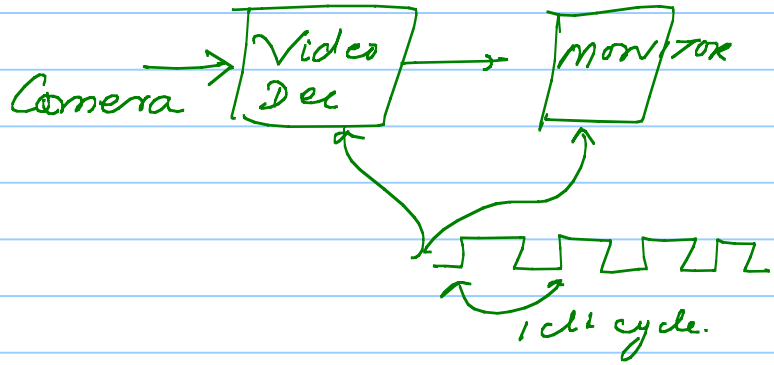
Asynchronous Circuit



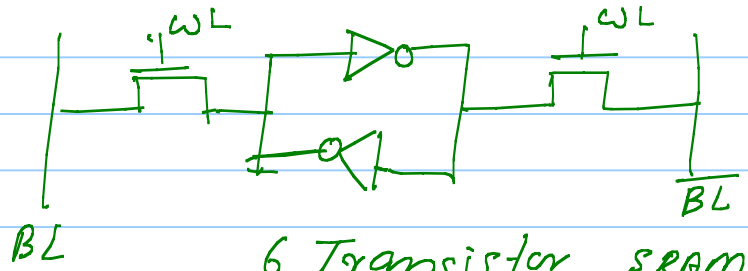
Synchronous Circuit

Some notion of timing

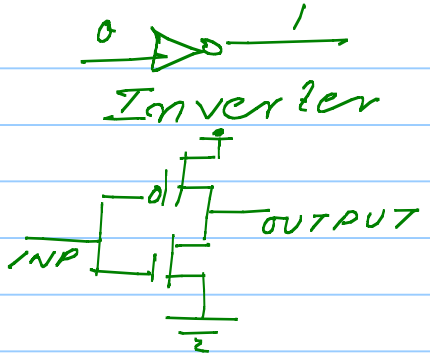




SRAM → Static Random Access Memory.



6 Transistor SRAM cell.



Simplify the big picture.

Architecture : What is visible to software/compiler.
- (i) Set of instruction.
- (ii) Set of registers (Explained later)
- (iii) Virtual view of memory.

Organization: Hardware sees everything.
- (i) caches.
- (ii) Registers. Program Counter Internal cpu state

Architecture — Software / Compiler's point of view

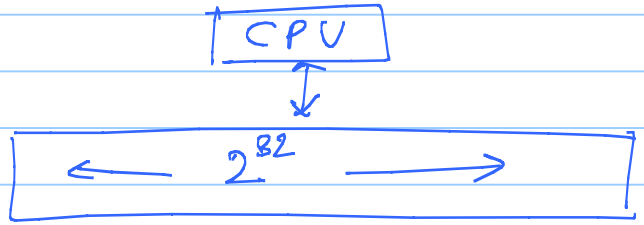
{black box}

32 bit architecture

↳ all my instructions are 32 bits long. (Not true always).

→ All my variables have a size of 32 bits.

↳ The maximum memory that I can access is 2^{32} bytes 4GB.



Illusion that all the memory is yours is called virtual memory

Virtual Memory is a good thing

→ it delinks the program from the architecture of the memory system.

→ compiler can freely generate code

Next Class:

1) Review
2) Registers
3) Assembly Programming