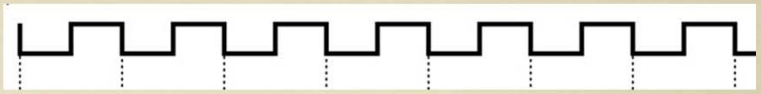
CPU Architecture

system clock



Memory

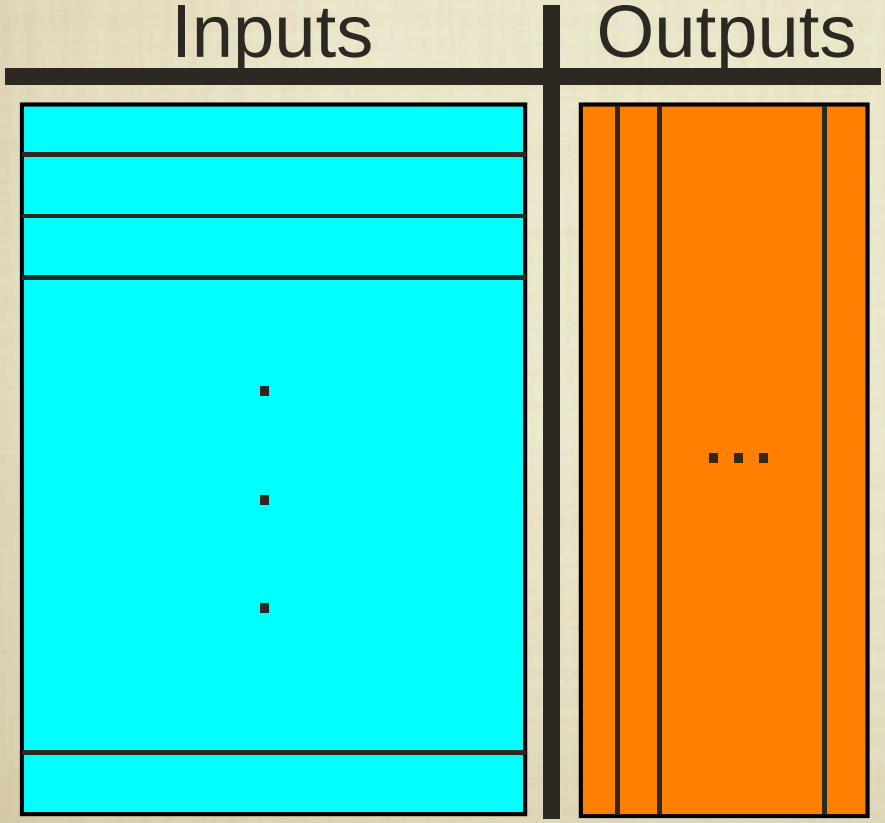
64-bit adder

Every CPU architecture is implemented using digital logic. In each cycle of the system clock, logic is “executed” and results are saved. System designers attempt to implement logic so that it can be executed in one clock cycle.

Circuits and Logic

- What kinds of functions can we actually implement as a digital circuit?

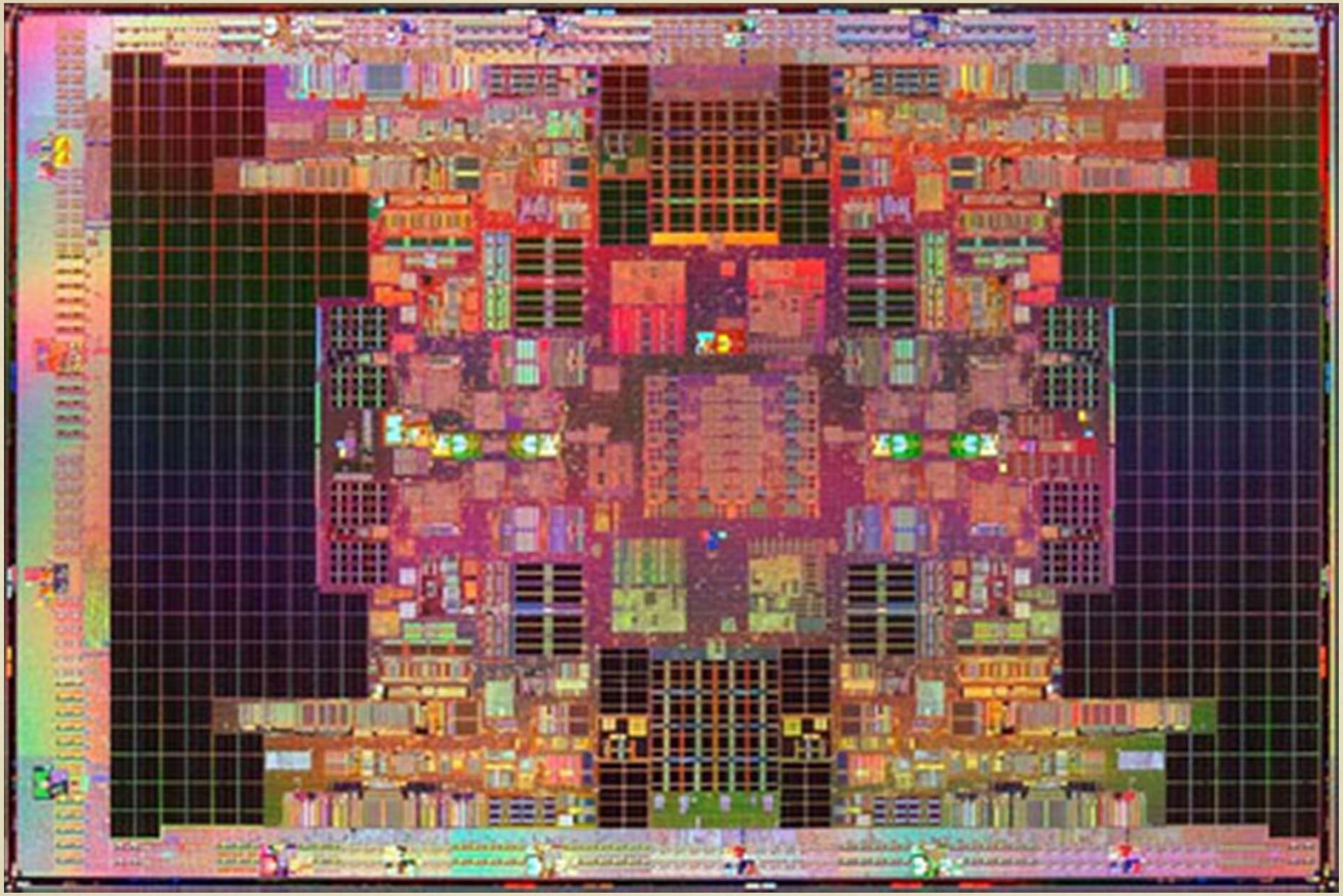
- Any “Boolean” function can be converted into a circuit:



We can construct a circuit for each output bit that is a function of the inputs.

The goal is to produce circuits as fast as possible - that is, with minimum “depth.”

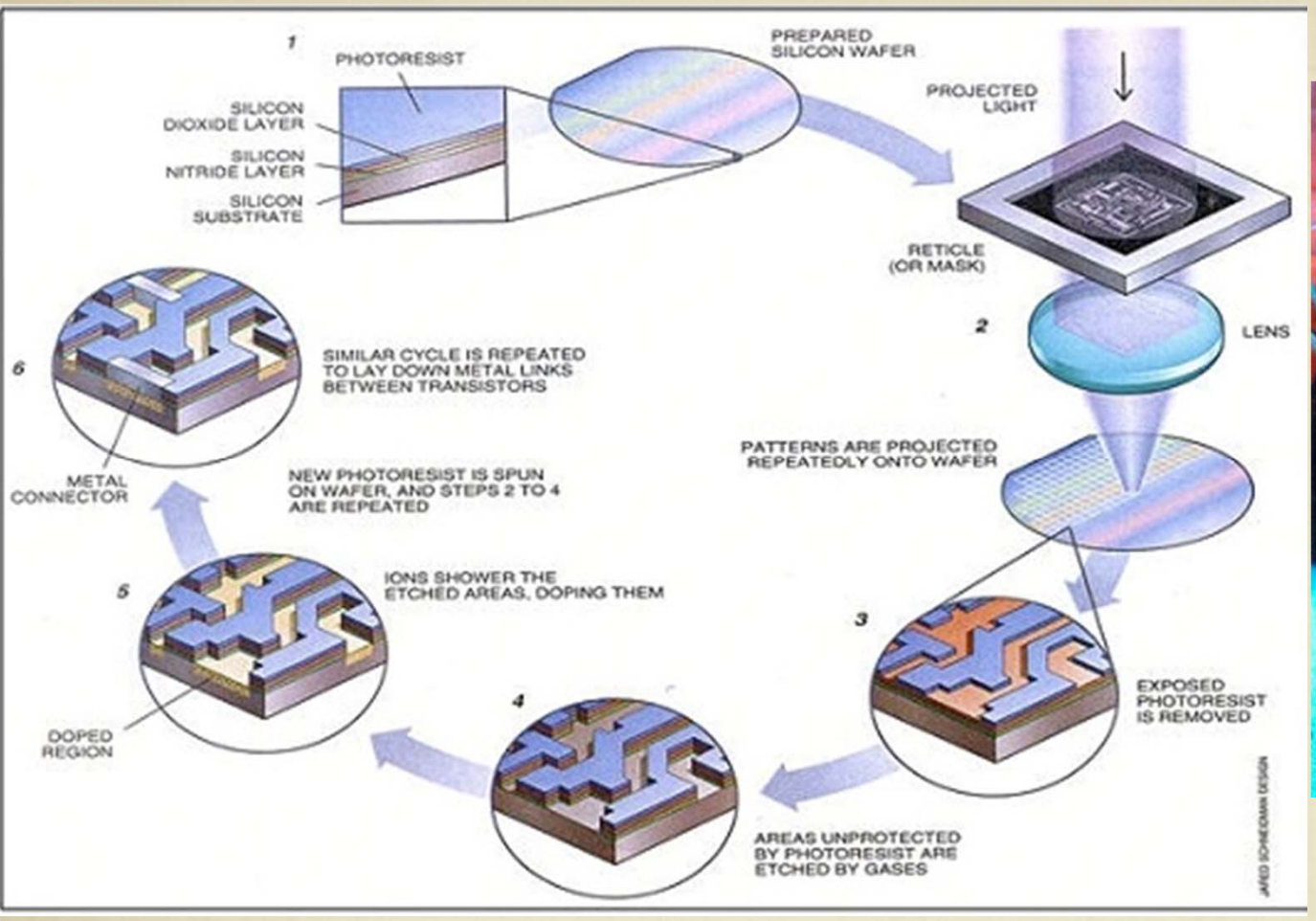
CPU Architecture

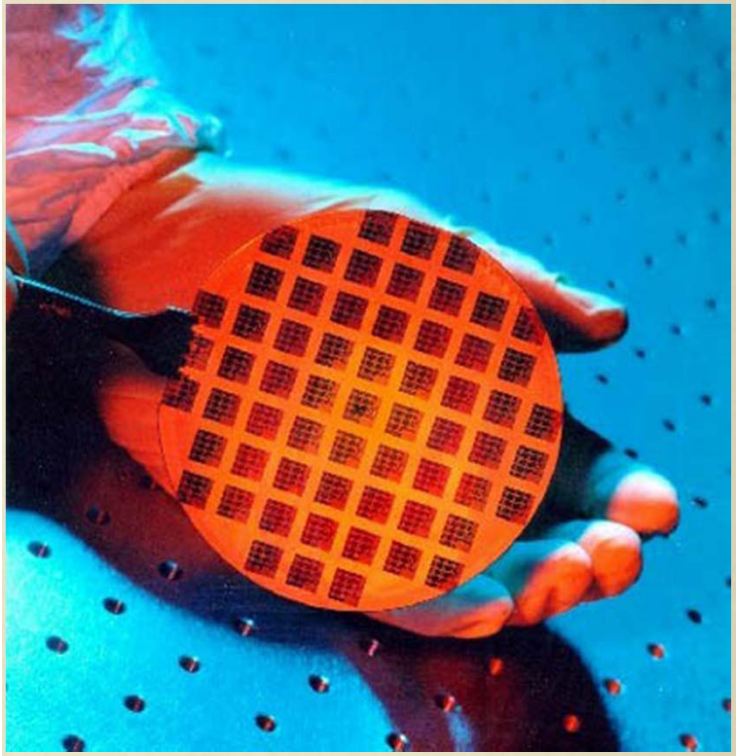


Intel Itanium (2001)

How is the logic created? How is silicon used?

Semiconductor Technology





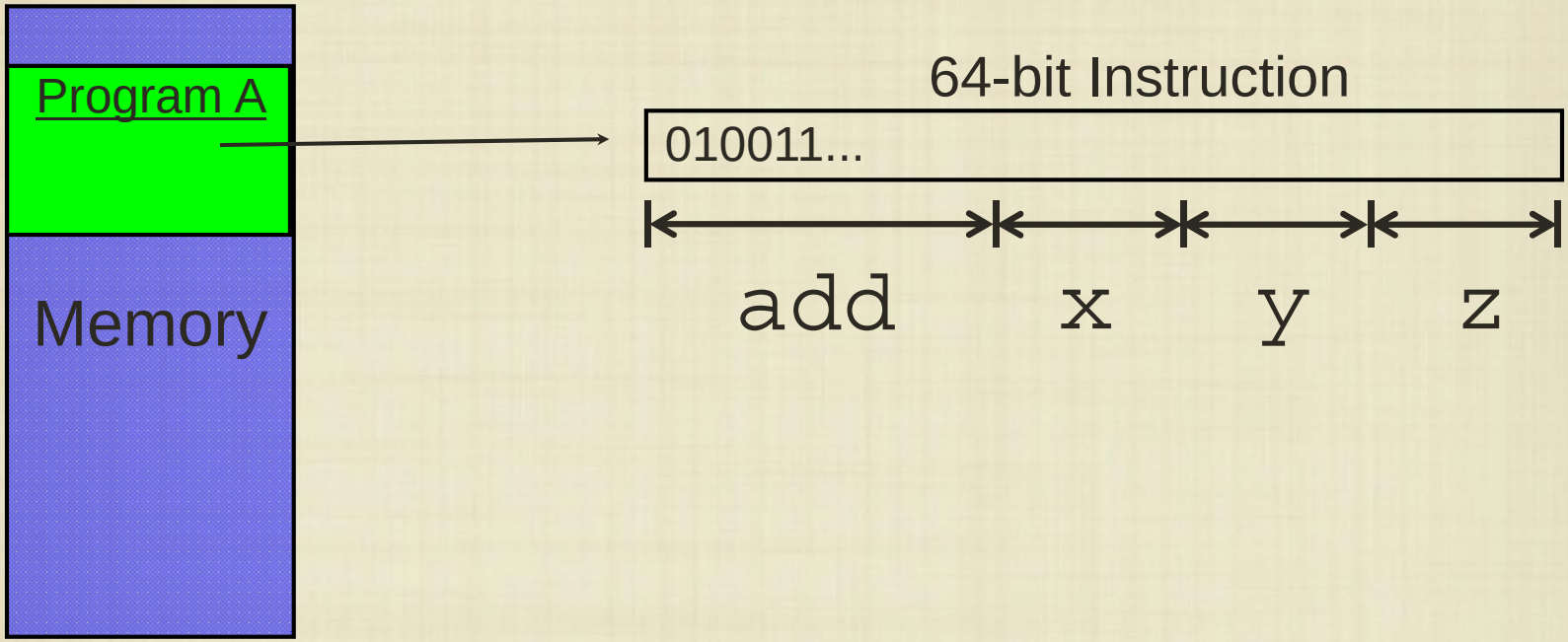
Semiconductor photolithography

Modern CPUs are constructed by etching transistors (and thus gates) into silicon. “Feature size” is on the order of tens of nanometers; CPUs can have several billion transistors.

Everything is a logic operation!

CPU Instruction Sets

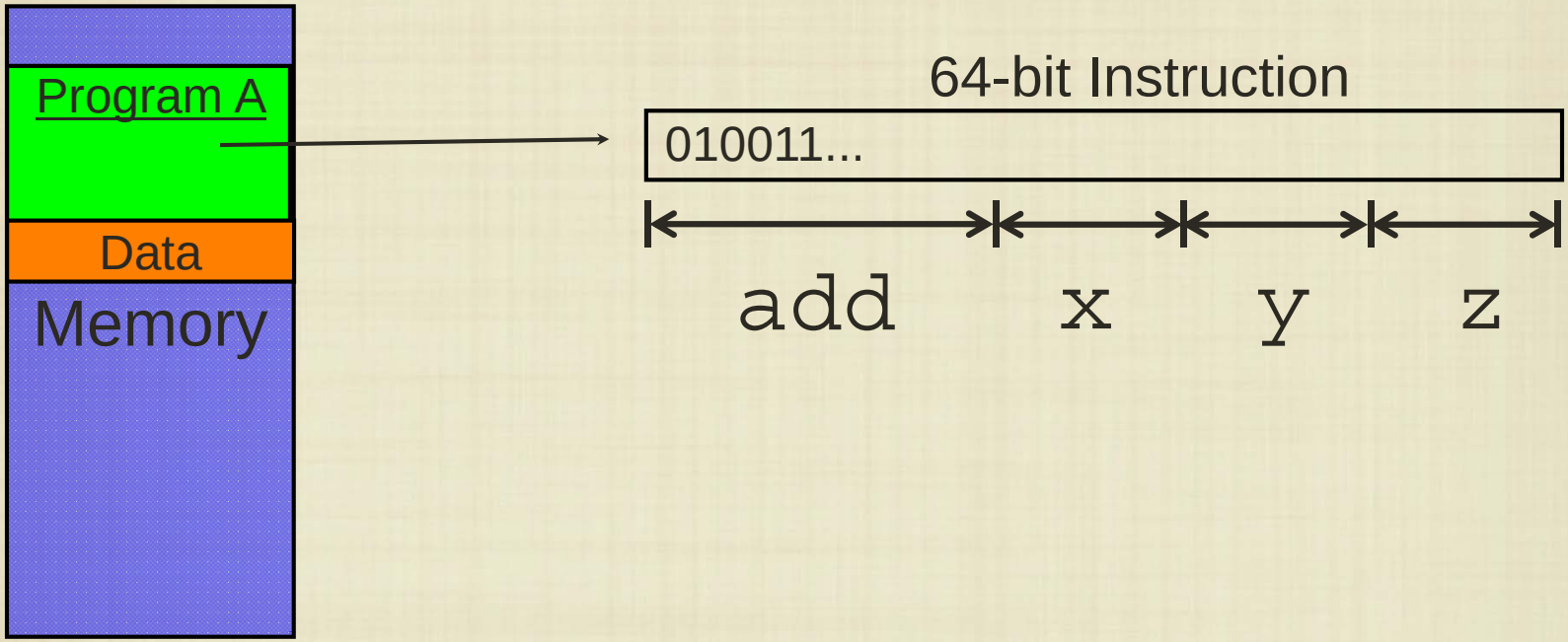
How are programs stored in memory? Programs are just a special kind of data that control the CPU.



Every CPU has a predefined instruction set that determines which circuitry is active. The main categories of instruction type are: memory access, control flow, and arithmetic operations.

CPU Instruction Sets

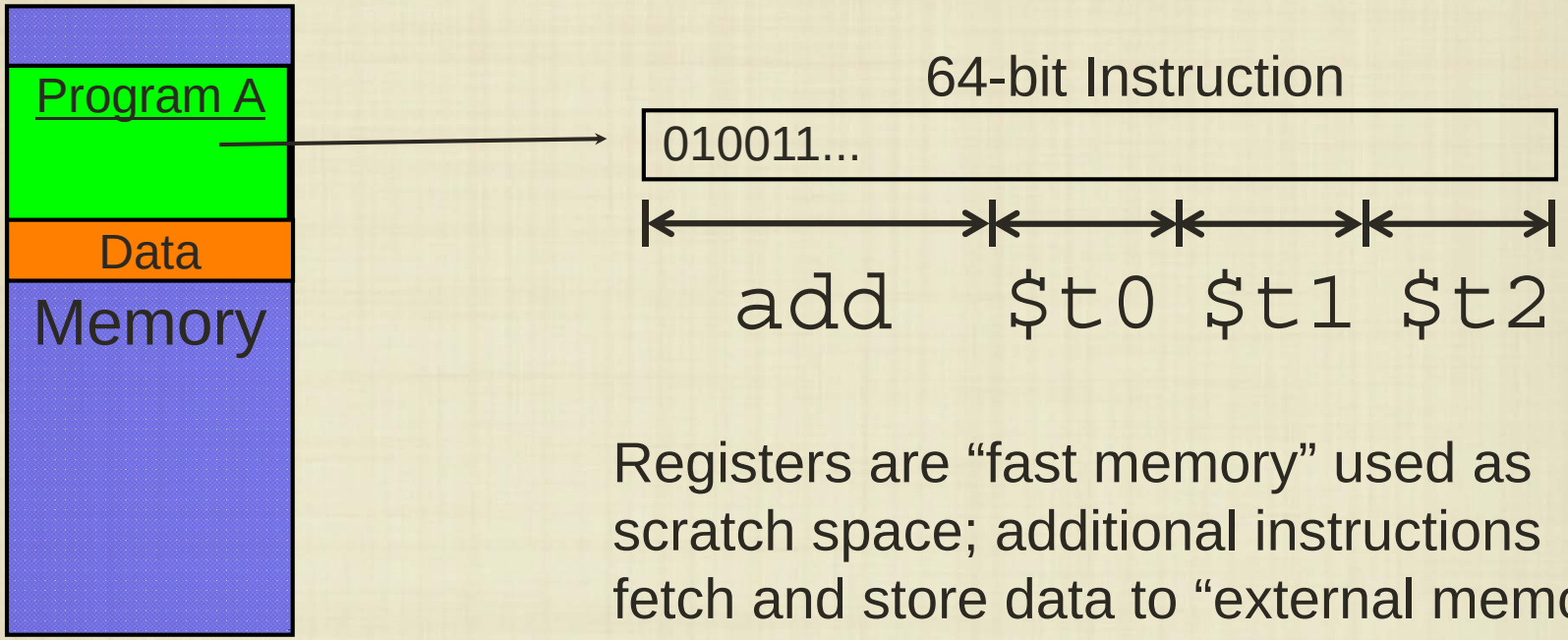
How are programs stored in memory? Programs are just a special kind of data that control the CPU.



Every CPU has a predefined instruction set that determines which circuitry is active. The main categories of instruction type are: memory access, control flow, and arithmetic operations.

CPU Instruction Sets

How are programs stored in memory? Programs are just a special kind of data that control the CPU.



Registers are “fast memory” used as scratch space; additional instructions fetch and store data to “external memory”.

Every CPU has a predefined instruction set that determines which circuitry is active. The main categories of instructions are: memory access, control flow, and arithmetic operations.

“Machine Language”

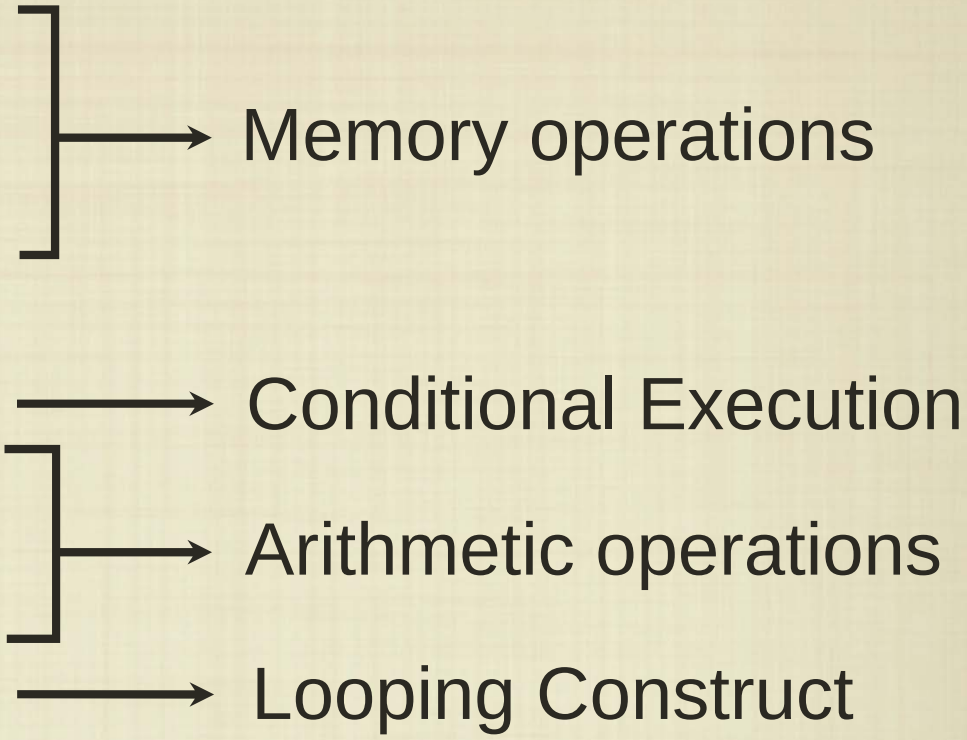
```
1  lw $t0, 1
2  lw $t1,0
3  lw $t2, n
4  loop:
   bgt $t0,$t2,done
5  add $t0, $t1, $t1
6  add $t0, 2
7  jmp loop
8  done:
```

`sw`: store word to register
`lw`: load word to register
`bgt`: branch-on-greater-than
`add`: add two registers
`jmp`: jump to given address

What does this program do?

“Machine Language”

```
1  lw $t0, 1
2  lw $t1,0
3  lw $t2, n
4  loop:
   bgt $t0,$t2,done
5  add $t0, $t1, $t1
6  add $t0, 2
7  jmp loop
8  done:
```



High-Level Constructs

Program A

```
z = x + y
    ...
if (a <= b)
    sum = sum + 1
else:
    sum = sum - 1
    ...
while (a <= b)
    sum = sum + 1
```

Arithmetic Operations

Conditional Execution

Looping constructs

You may be more familiar with the above constructs - how do they compare with machine instructions?

Higher-Level Languages

Machine Language

```
lw $t0, 1
lw $t1,0
lw $t2, n
loop:
bgt $t0,$t2,done
add $t0, $t1, $t1
add $t0, 2
jmp loop
done:
```

C/C++

```
int sum = 0;
for (int i=1; i<=n; i+=2)
    sum += i;
```

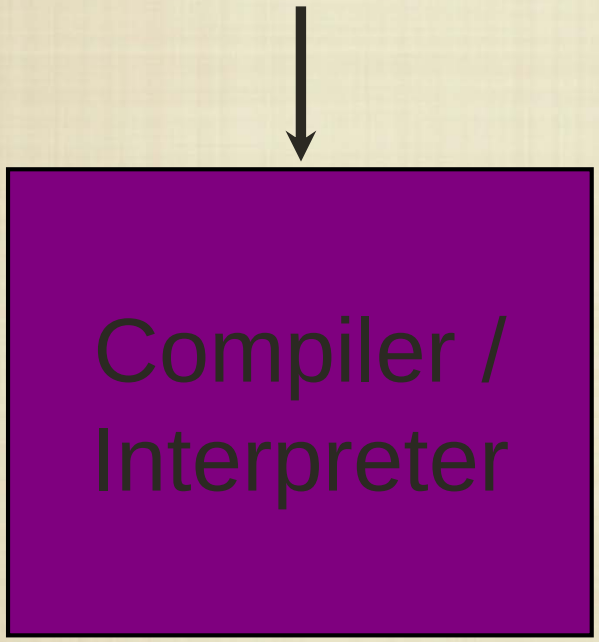
Python

```
sum = 0
i = 1
while (i <= n):
    sum += i
    i += 2
```

High-level programming languages allow us to forget the painful details of machine instructions, so that we can develop concise ways to perform useful tasks.

Creating Machine Instructions

```
sum = 0
i = 1
while (i <= n):
sum += i
i += 2
```




```
lw $t0, 1
lw $t1,0
lw $t2, n
loop:
bgt $t0,$t2,done
add $t0, $t1, $t1
add $t0, 2
jmp loop
done:
```

Compilers are CPU-specific programs that translate from high-level language to machine code.

Recap

- What architecture does every computational device have? What are the features of this architecture?
- Is a hard drive an input device or output device?
- What does a program consist of and where is it stored? How are machine instructions executed on the CPU?
- What is a logic gate? What are the three basic logic operations?
- What CPU operations are implemented using digital logic?
- What is a program? How is it executed on the CPU?