CO200 – Computer Organization and Architecture

Basavaraj Talawar, CSE, NITK

Learning from the Course

- How does the hardware execute our program?
  - What goes on ‘under the hood’ during program execution?

Learning from the Course

- How does the hardware execute our program?
  - What goes on ‘under the hood’ during program execution?
- Which components of the system are ‘at work’?
  - Design of these components

What does the hardware see?

```
int main() {
 int i, a=2, b=3, sum=0;
 sum = a + b;
}
```

What the Programmer sees.

What does the hardware see?

```
int main() {
 int i, a=2, b=3, sum=0;
 sum = a + b;
}
```

What the Programmer sees.

Compiler

What does the hardware see?

```
int main() {
 int i, a=2, b=3, sum=0;
 sum = a + b;
}
```

What the Programmer sees.

Compiler

Binary Code

```
00000000101000100000000100011000
00000000100000100001000000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
00000011111000000000000000001000
```

What the Machine sees.

What does the hardware see?

Binary Code

```
00000000101000100000000100011000
00000000100000100001000000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
00000011111000000000000000001000
```

- What does the binary code (`a.out`) contain?
- What happens when do `$ ./a.out`

The Program

```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Start here!

The Program

```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Allocate space in memory for these data!

The Program

```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

The Program

```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Exit the program!

What goes on under the hood?

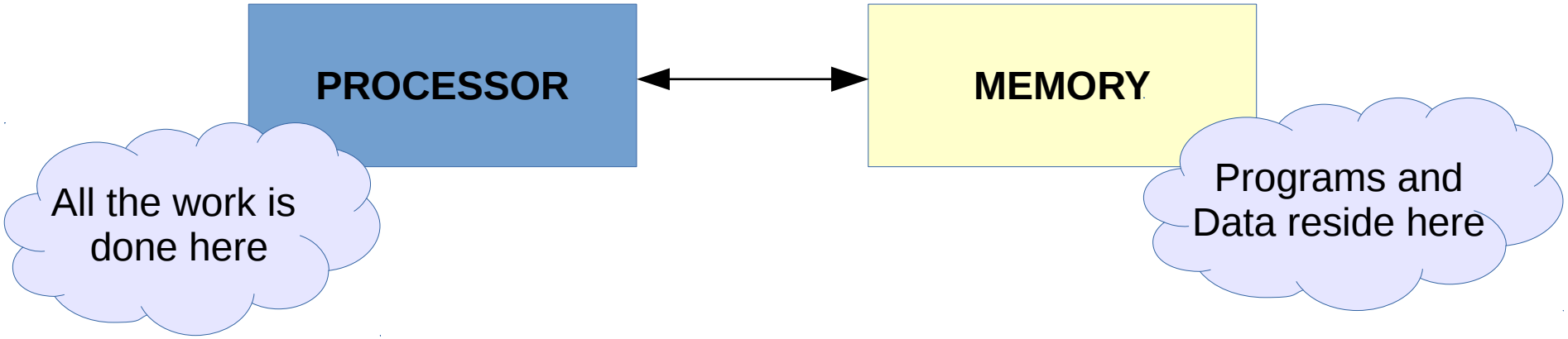
```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

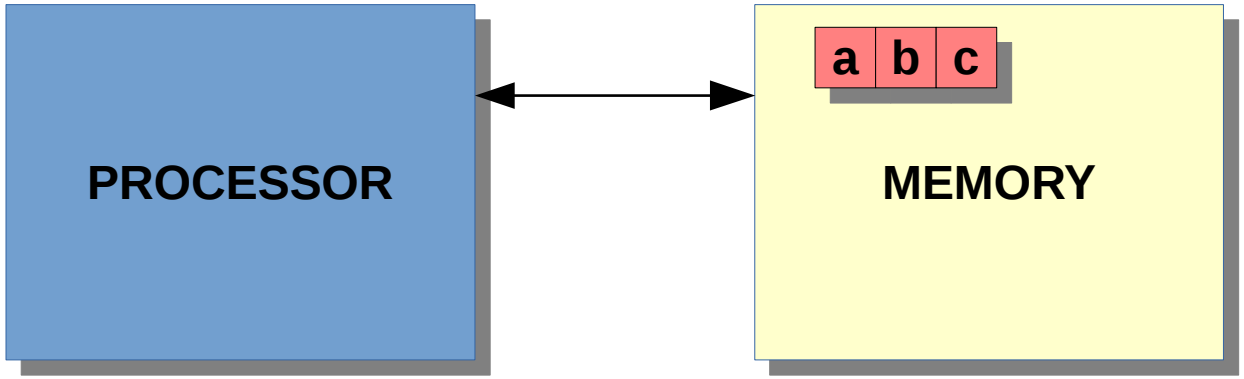
What goes on under the hood?

```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.



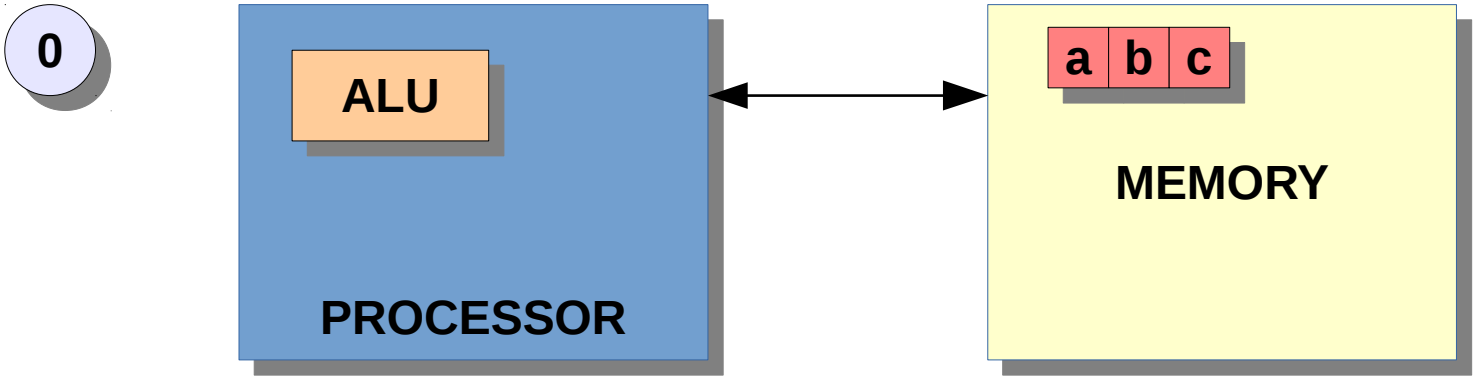
What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

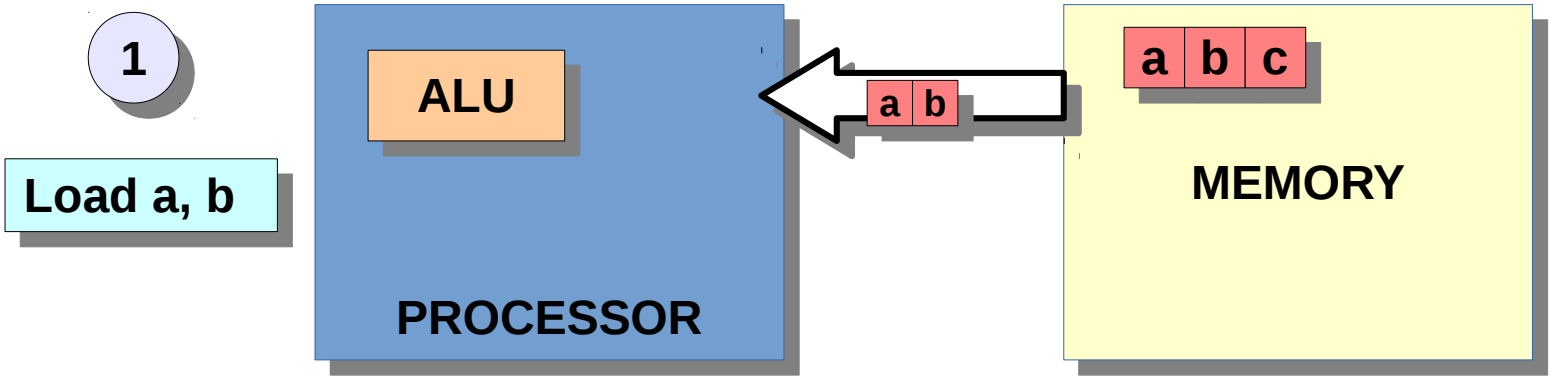
What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

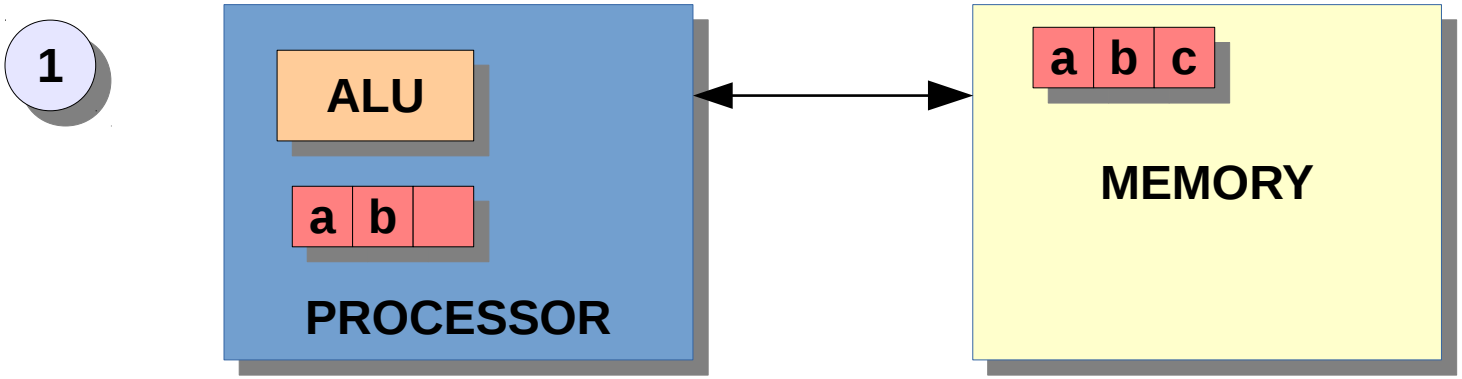
What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

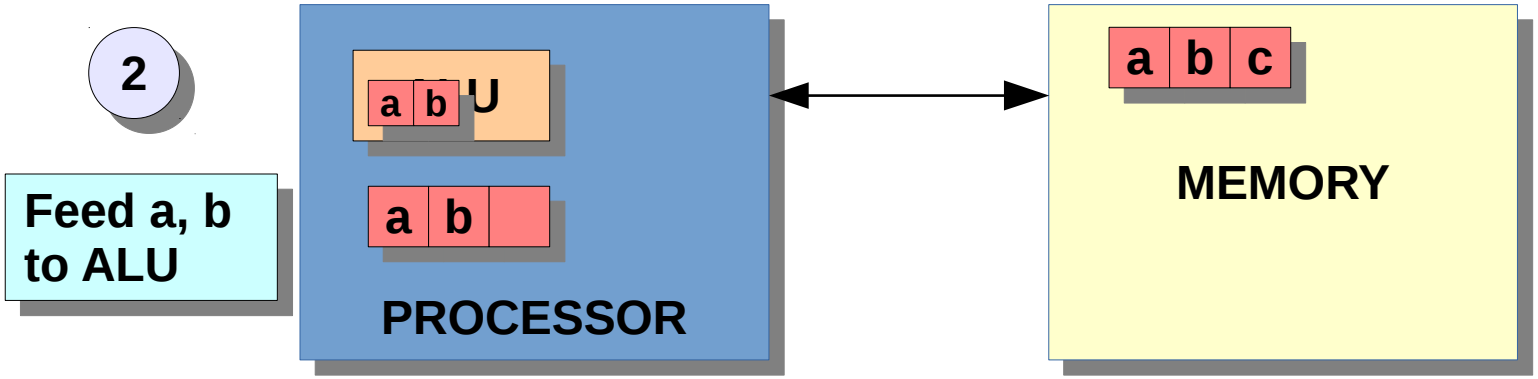
What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

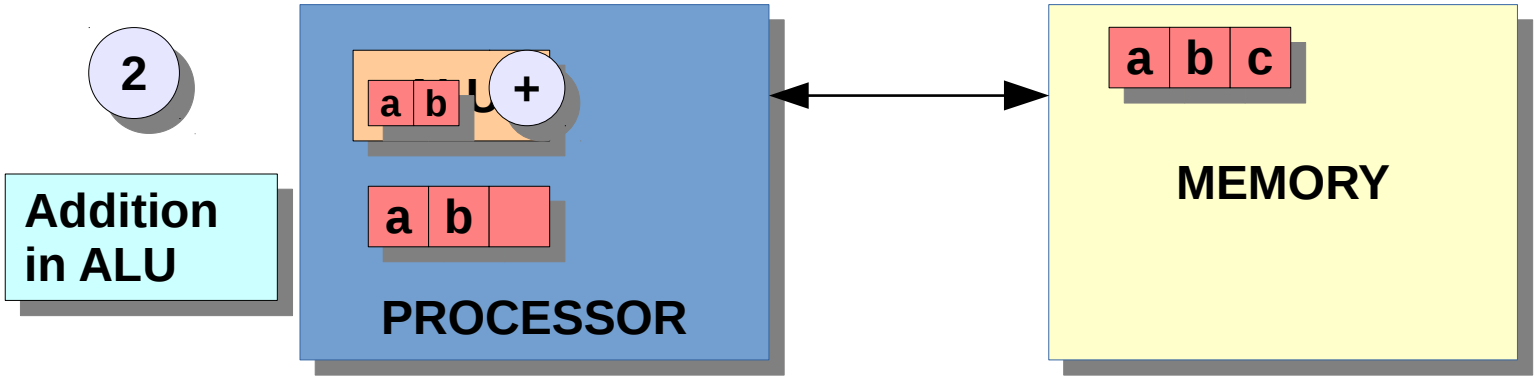
What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

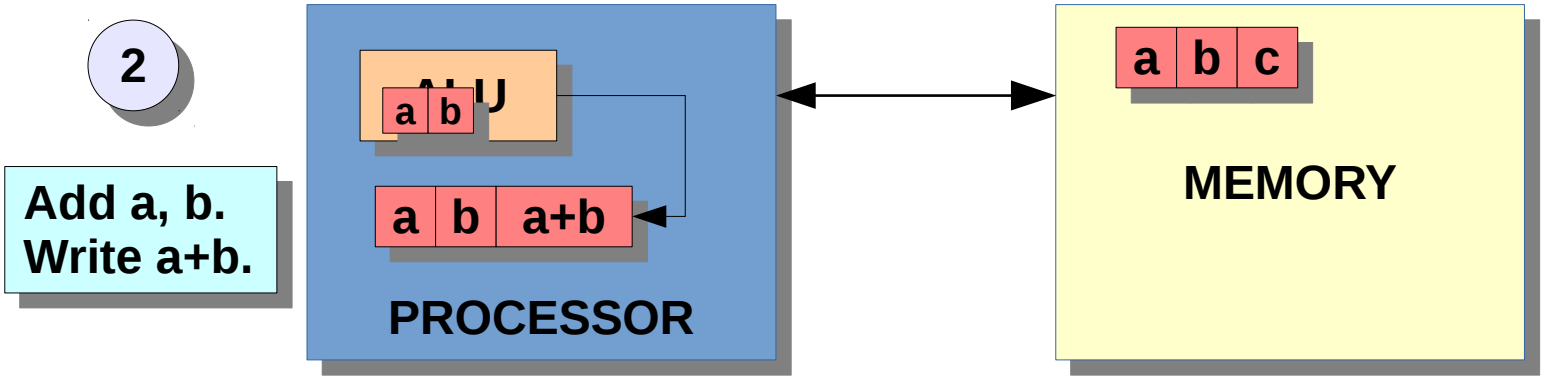
What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

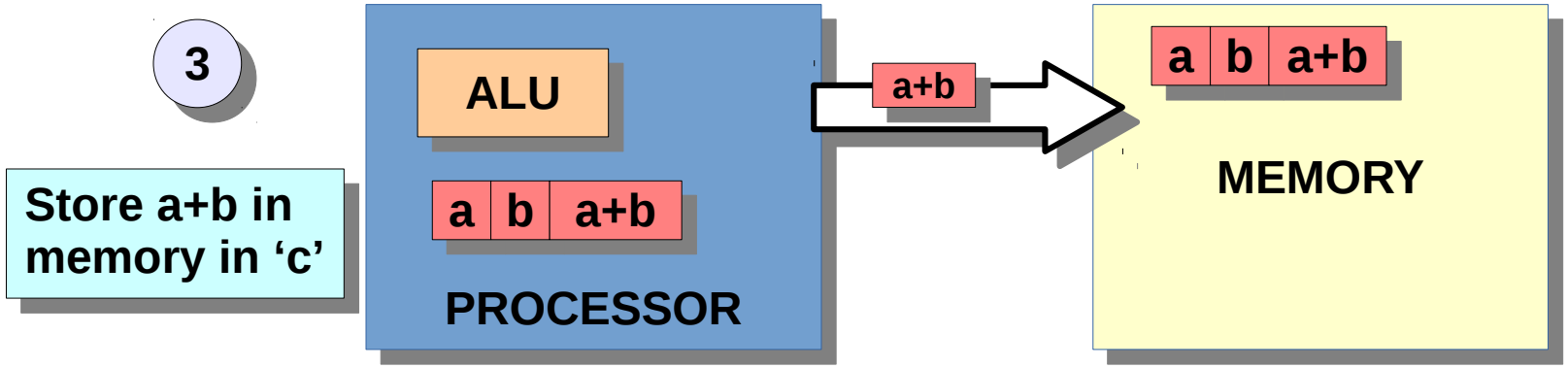
What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

What goes on under the hood?




```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

What goes on under the hood?

```
int main() {
 int a=2, b=3, sum=0;
 sum = a + b;
}
```

Add a and b and store the result in c.

- Variables – memory locations
- Load the values from memory into the processor
- Feed the inputs to the ALU
- Arithmetic operation in the ALU – Addition
- Save the sum in the processor
- Store the calculated sum from the processor to the memory

A Bigger Program

```
int calculate_sum(int a[], int i) {
 int sum=0;
 i=0;
 for(i=0;i<5;i++)
   sum = sum + a[i];
return sum;
}

int main() {
 int i, a[5]={2,3,5,7,11}, sum=0;
 sum=calculate_sum(a, 5);
}
```

A Bigger Program

```
int calculate_sum(int a[], int i) {
 int sum=0;
 i=0;
 for(i=0;i<5;i++)
   sum = sum + a[i];
return sum;
}

int main() {
 int i, a[5]={2,3,5,7,11}, sum=0;
 sum=calculate_sum(a, 5);
}
```

- Condition evaluation
- Function call and return
- Parameters pass and return

An Even Bigger Program !

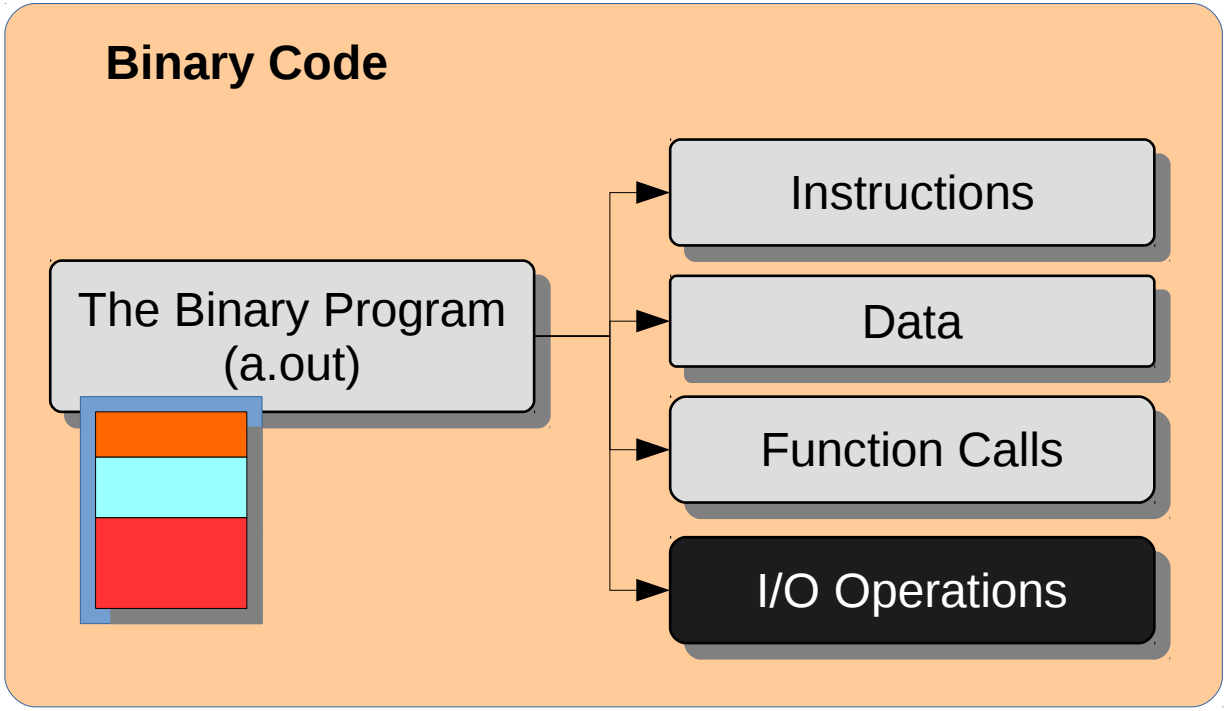
```
int calculate_sum(int a[], int i) {
 int sum=0;
 i=0;
 for(i=0;i<5;i++)
   sum = sum + a[i];
return sum;
}

int main() {
 int i, a[5]={2,3,5,7,11}, sum=0;
 sum=calculate_sum(a, 5);
 printf("the sum: %d.", sum);
}
```

An Even Bigger Program !

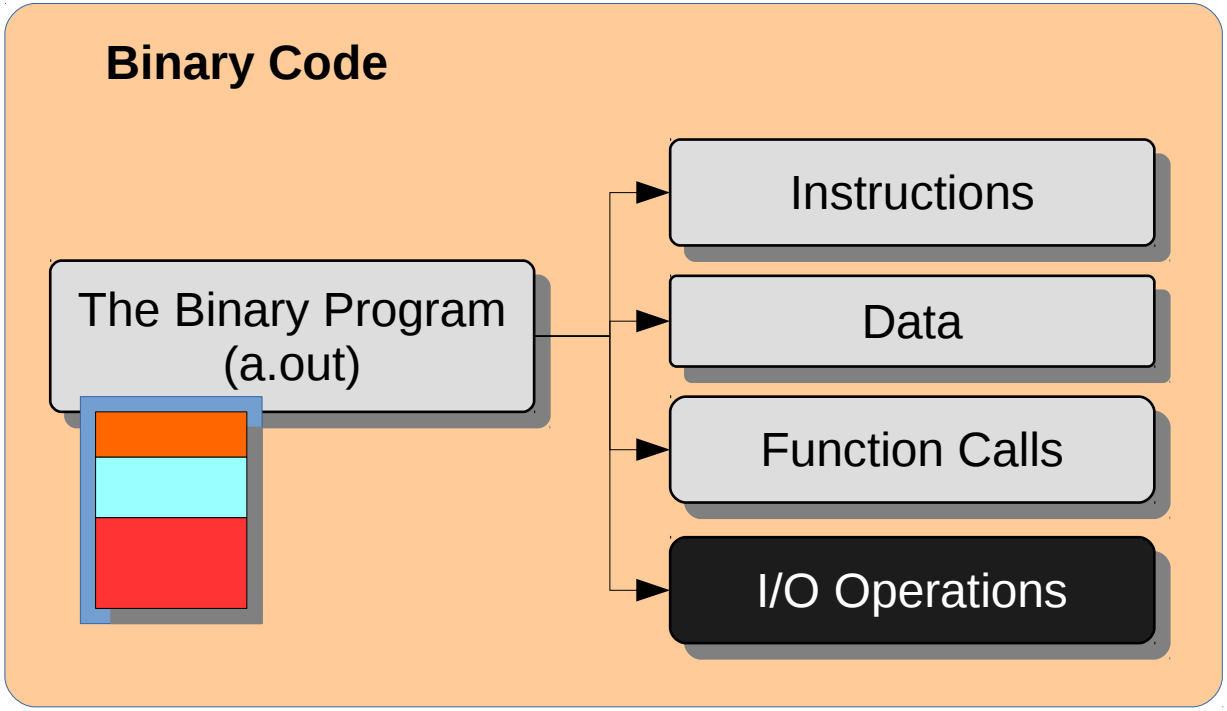
```
int calculate_sum(int a[], int i) {
 int sum=0;
 i=0;
 for(i=0;i<5;i++)
   sum = sum + a[i];
return sum;
}

int main() {
 int i, a[5]={2,3,5,7,11}, sum=0;
 sum=calculate_sum(a, 5);
 printf("the sum: %d.", sum);
}
```

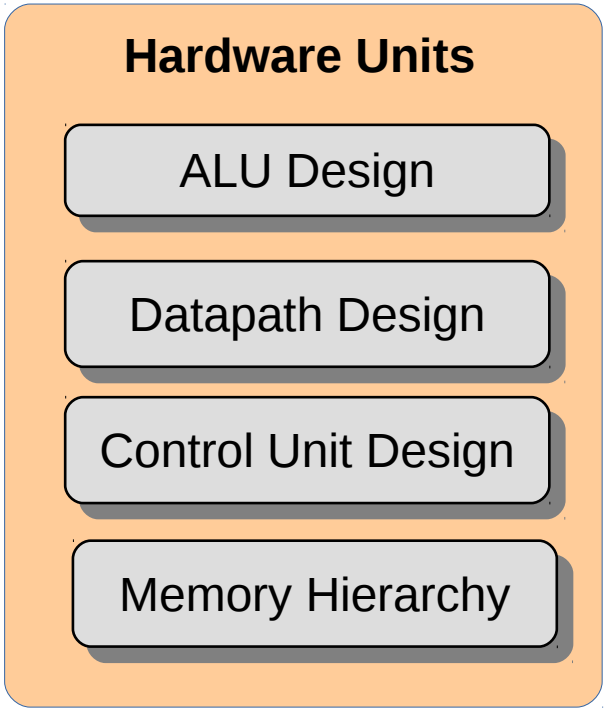
- I/O operation !

Computer Organization and Architecture – This Course



Computer Organization and Architecture – This Course





Course Details

- Assignments (~4)
  - MIPS Assembly language programming
  - Hardware design - Build components in SystemC
- Tutorials (8 – 10)
  - Solve problems in class; Teams of 2; Every week.
- Quizzes (2), Midterm and Final Exam
- Class slides will be on the course website

Course Reference Texts

- **David A Patterson and John L Hennessy. Computer Organization and Design – The Hardware/Software Interface. 5e, Morgan Kaufmann. 2014.**
- Hamacher, Vranesic, Zaky. Computer Organization, 5e. Tata McGraw Hill, 2011.
- John P Hayes. Computer Architecture and Organization, 3e. McGraw Hill, 1998.
- M. Morris Mano. Computer System Architecture. 3e. Pearson, 2007.
- David Harris and Sarah Harris. Digital Design and Computer Architecture. 2e. MK. 2013.
- NPTEL Courses (www.nptel.co.in)
  - Matthew Jacob – High Performance Computing, Bhaskaran Raman – Computer Organisation and Architecture, S. Raman – Computer Organization, Jatindra Kumar Deka – Computer Organisation and Architecture.