

# Week 12

## CS110P.07

Suting Chen

# Thread-Level Parallelism

- Basic concepts
- Multithreading for programmers
- Multithreading issue

Suting Chen

# Basic Concepts

- Processor
- Core
- Hyper-threading

Suting Chen

# Basic Concepts

## Processor

- Some physical stuff
- Can have more than 1

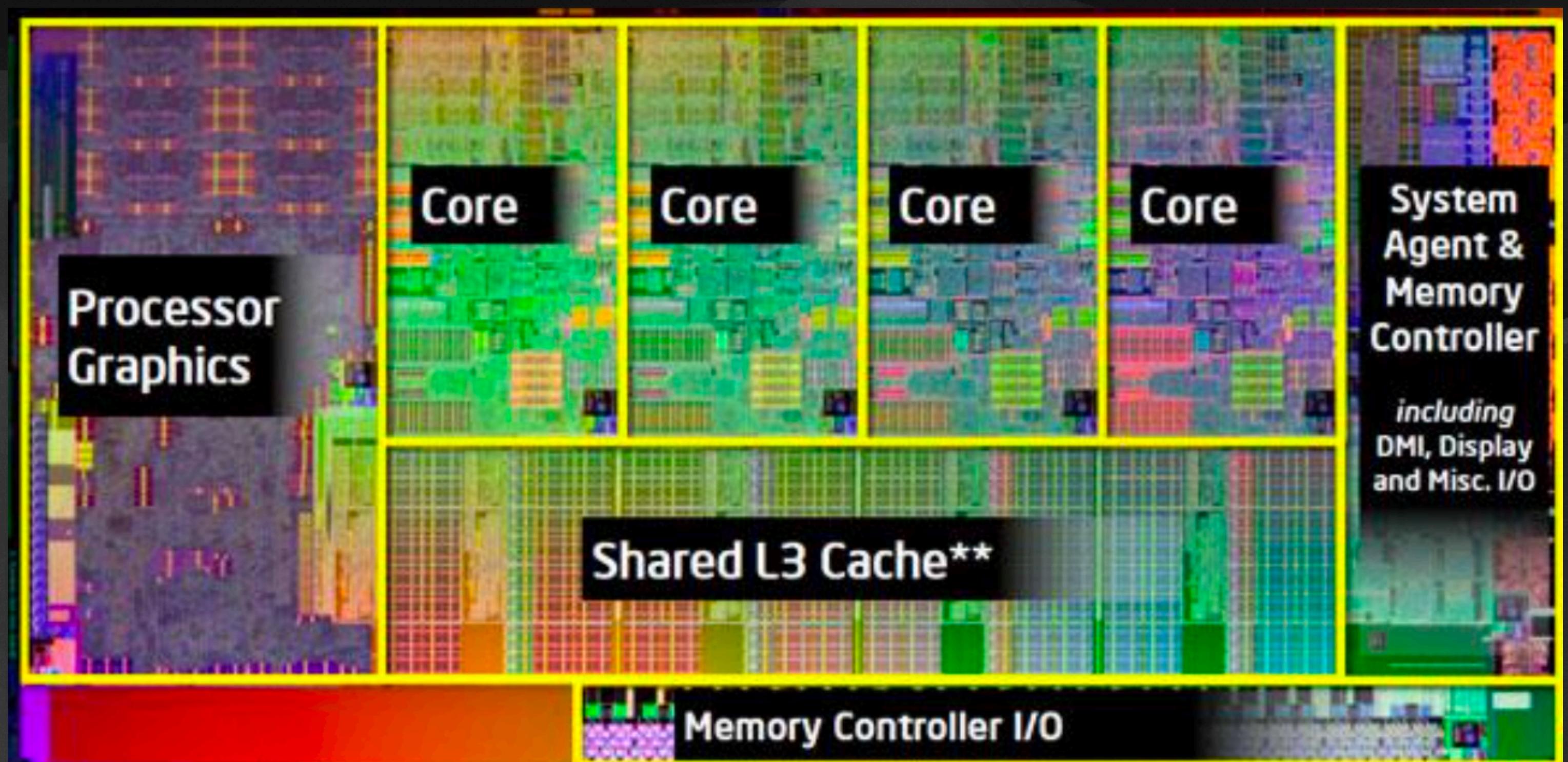


Suting Chen

# Basic Concepts

## Core

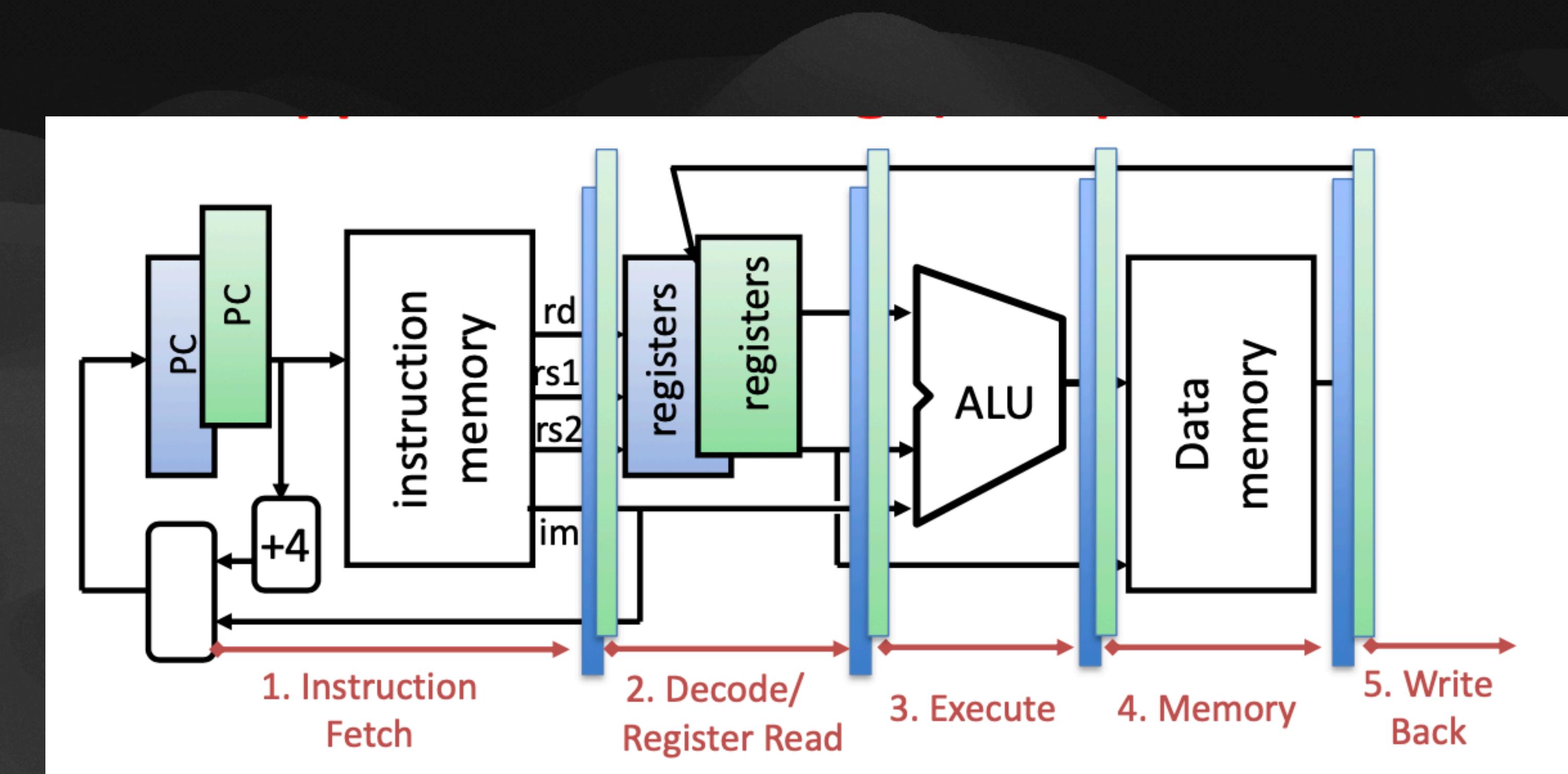
- Multiple cores in CPU
- Each core has:
  - PC
  - Register
  - ALU, FPU ...



# Basic Concepts

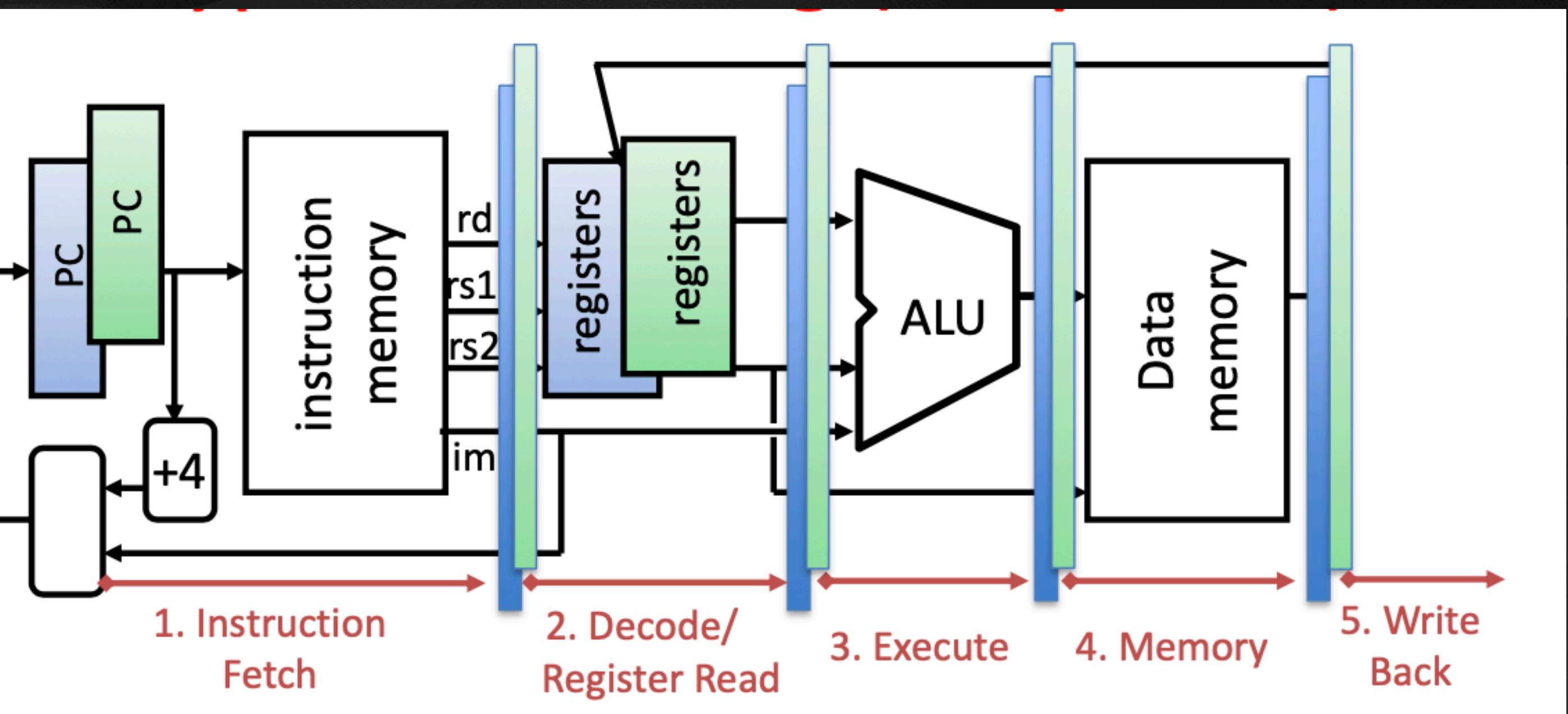
## Hyper-threading

- No calculation speed up
- Switch to another thread when waiting (for memory)



# Basic Concepts

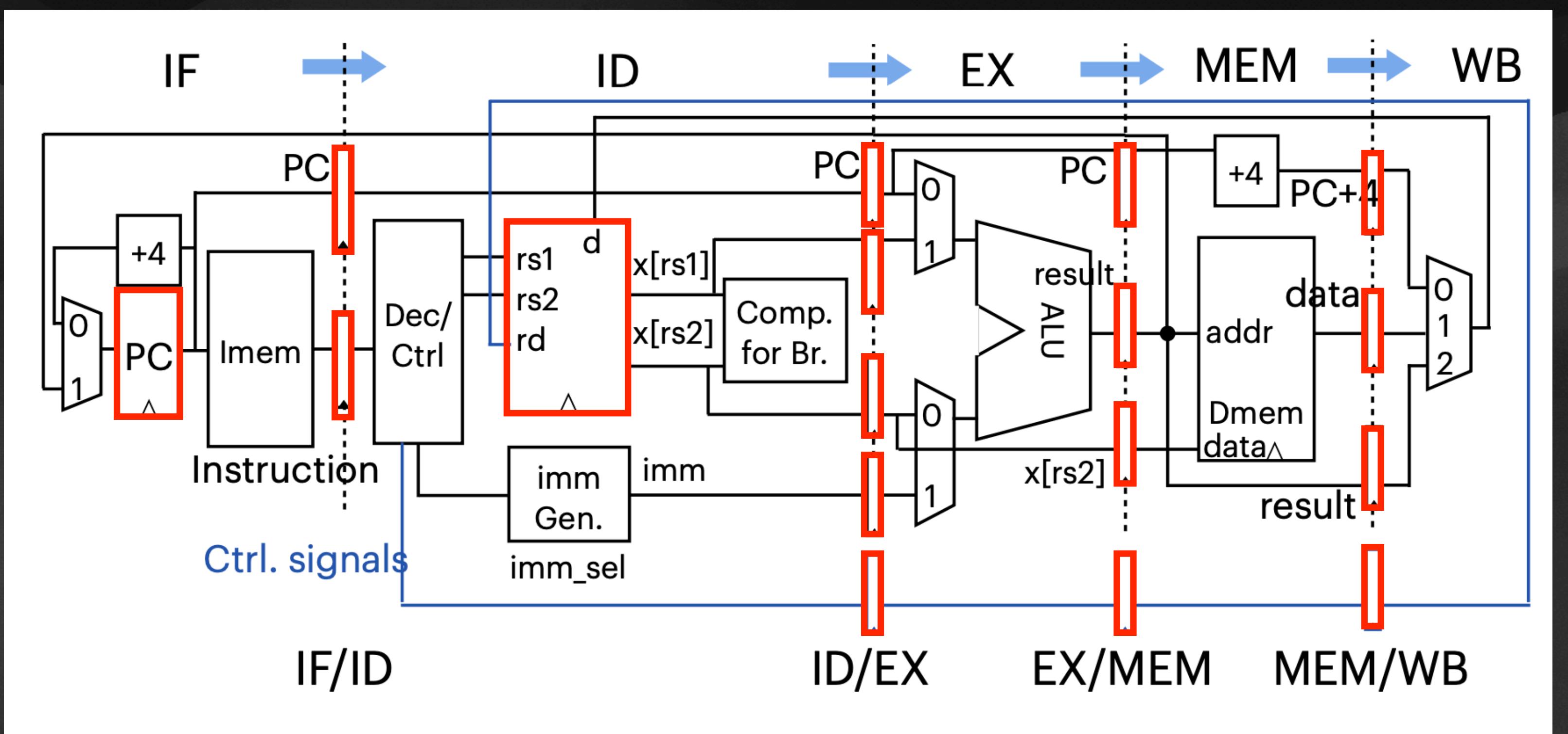
## What is a “thread”



```
for (int i = 0; i < 100; i++) {  
    pthread_create(&thread_id[i], NULL,  
                  hello, NULL);  
}
```

# Basic Concepts

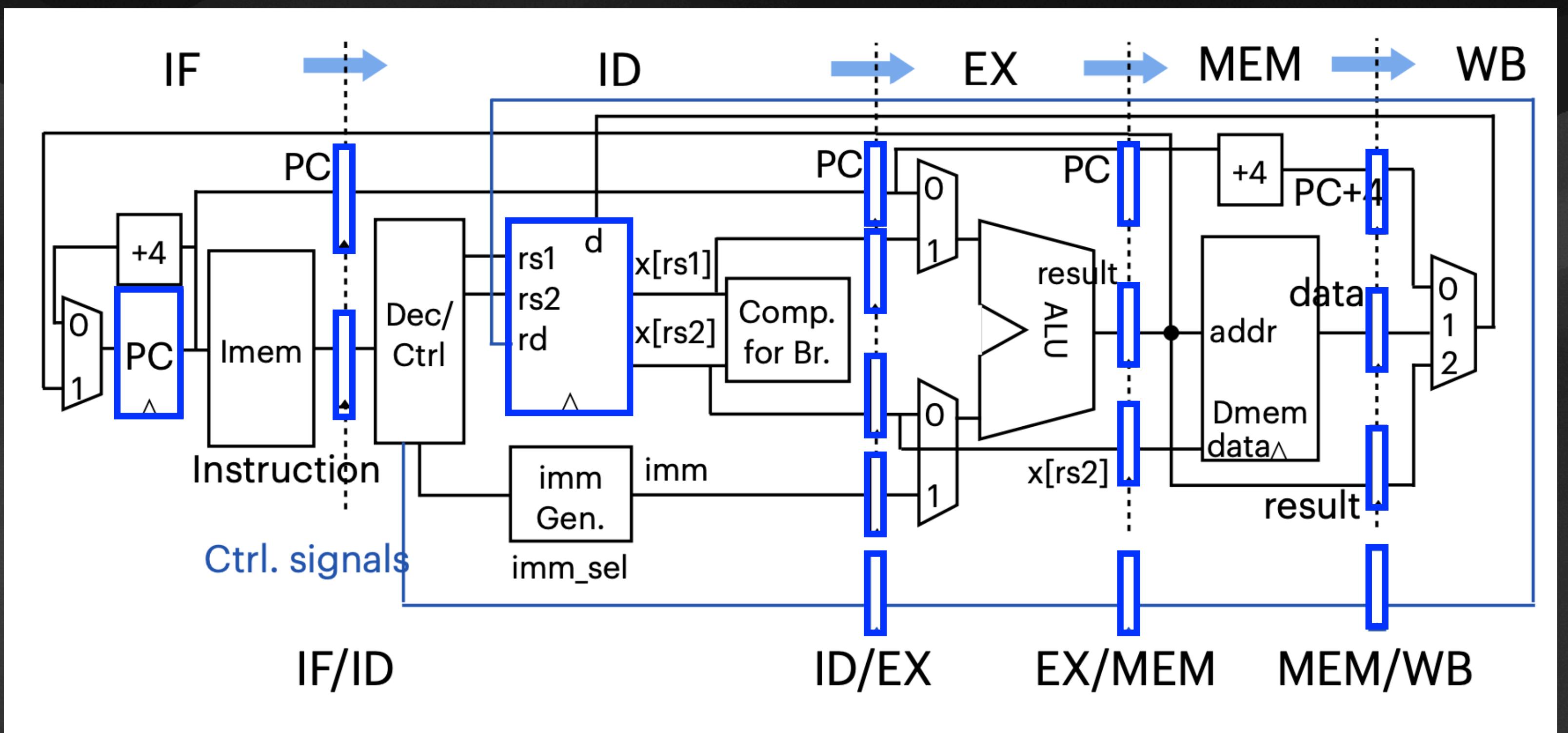
# What is a “thread”



Thread #	Information
143	

# Basic Concepts

What is a “thread”

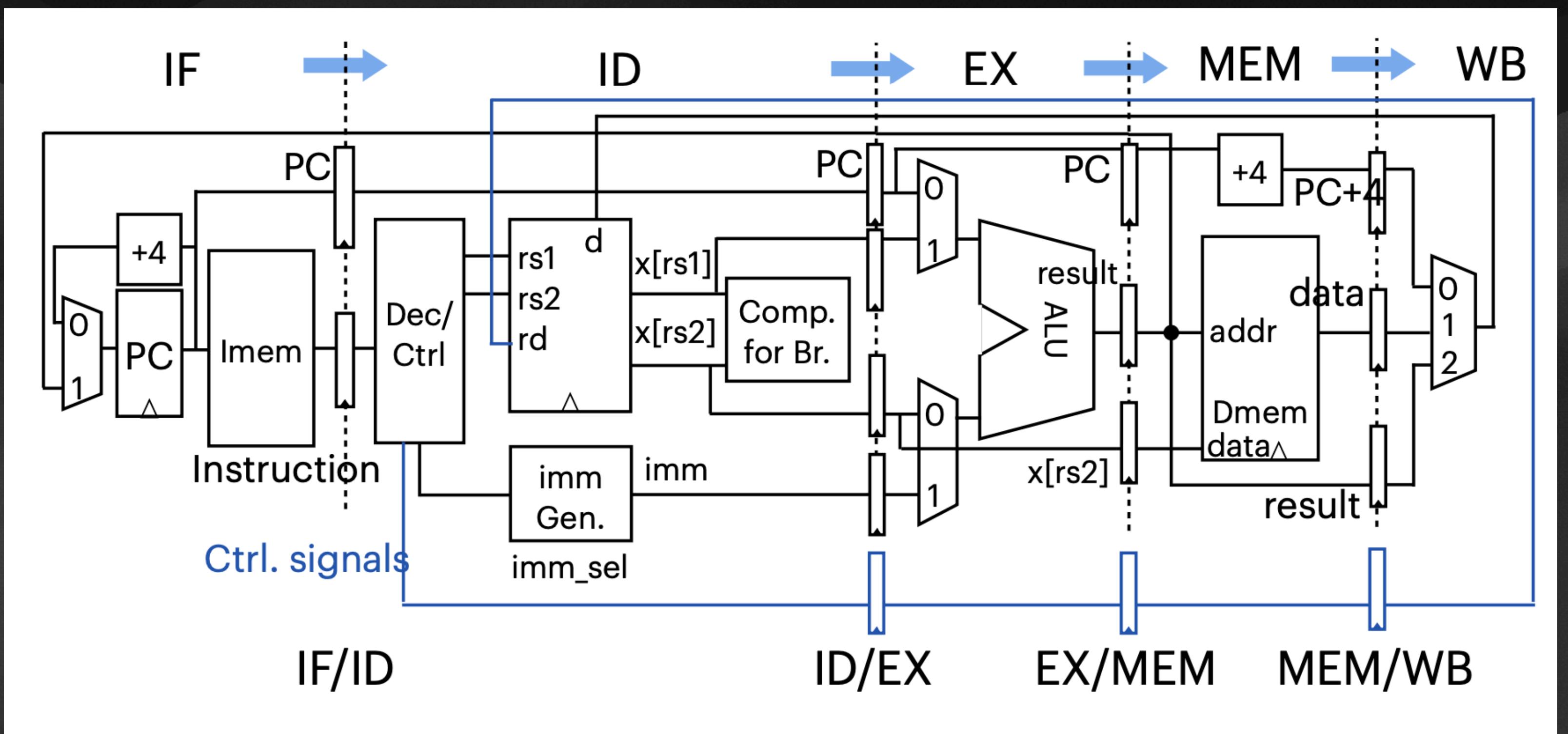


Thread #	Information
143	0   0   1   1   1   1
144	1   1   1   1   1   1

Suting Chen

# Basic Concepts

What is a “thread”

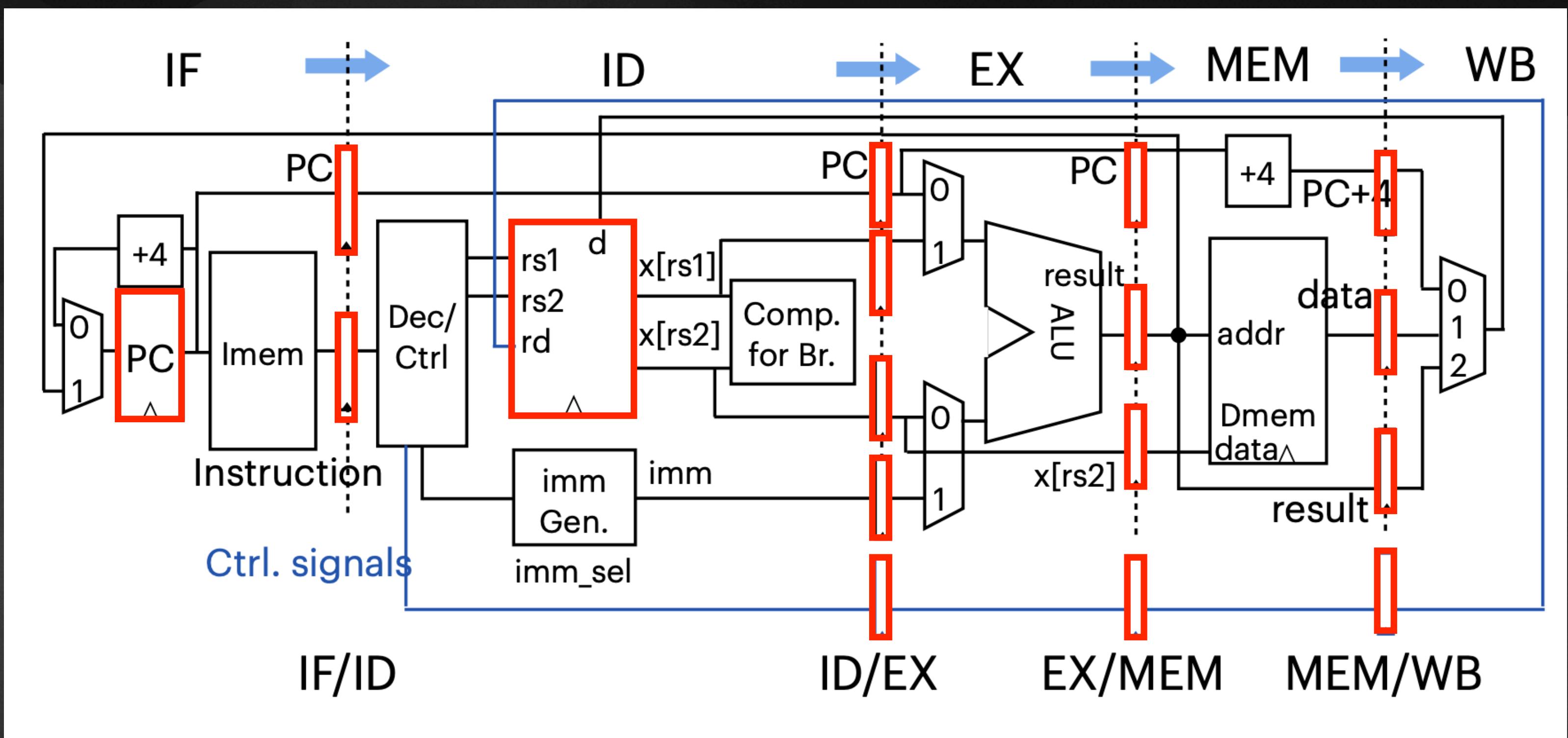


Suting Chen

Thread #	Information
143	Red dashed boxes
144	Blue dashed boxes

# Basic Concepts

What is a “thread”



Thread #	Information
143	
144	

# Context switch

- Switch from one task (thread) to another
- **LESS** effective than hyper-threading
- OS will help you!

# Multithreading for programmers

- pthread
- OpenMP
- This will be covered in next week discussion!



Suting Chen

# Multithreading issue

- Data race & Lock
- Design a lock
- Peterson algorithm (optional)

Suting Chen

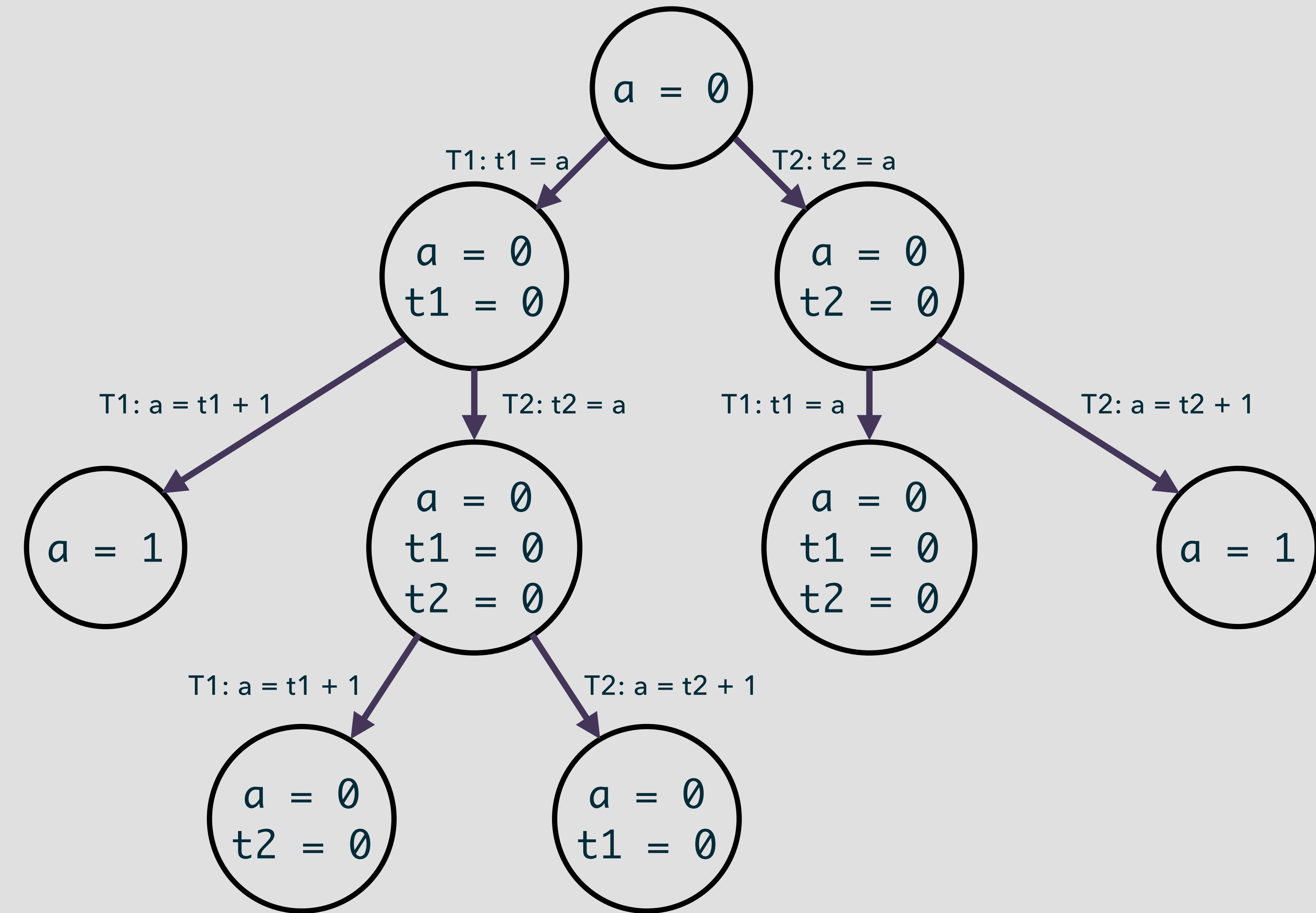
# NAIVE “ALIPAY”

```
a = 0

def T1():
    global a
    while True:
        t1 = a
        a = t1 + 1

def T2():
    global a
    while True:
        t2 = a
        a = t2 + 1
```

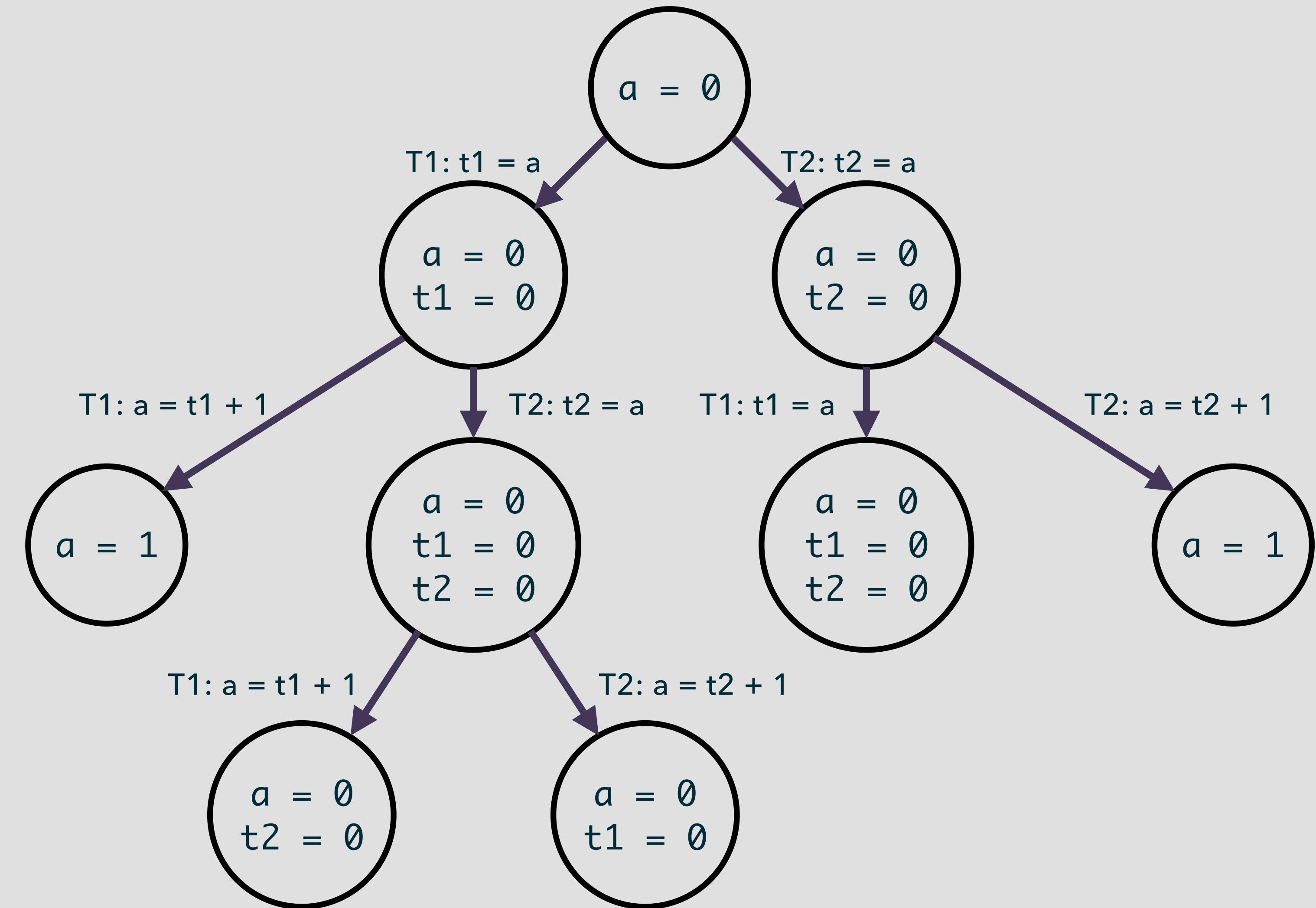
```
a = 0  
  
def T1():  
    global a  
    while True:  
        t1 = a  
        a = t1 + 1  
  
def T2():  
    global a  
    while True:  
        t2 = a  
        a = t2 + 1
```



```
a = 0

def T1():
    global a
    while True:
        t1 = a
        a = t1 + 1

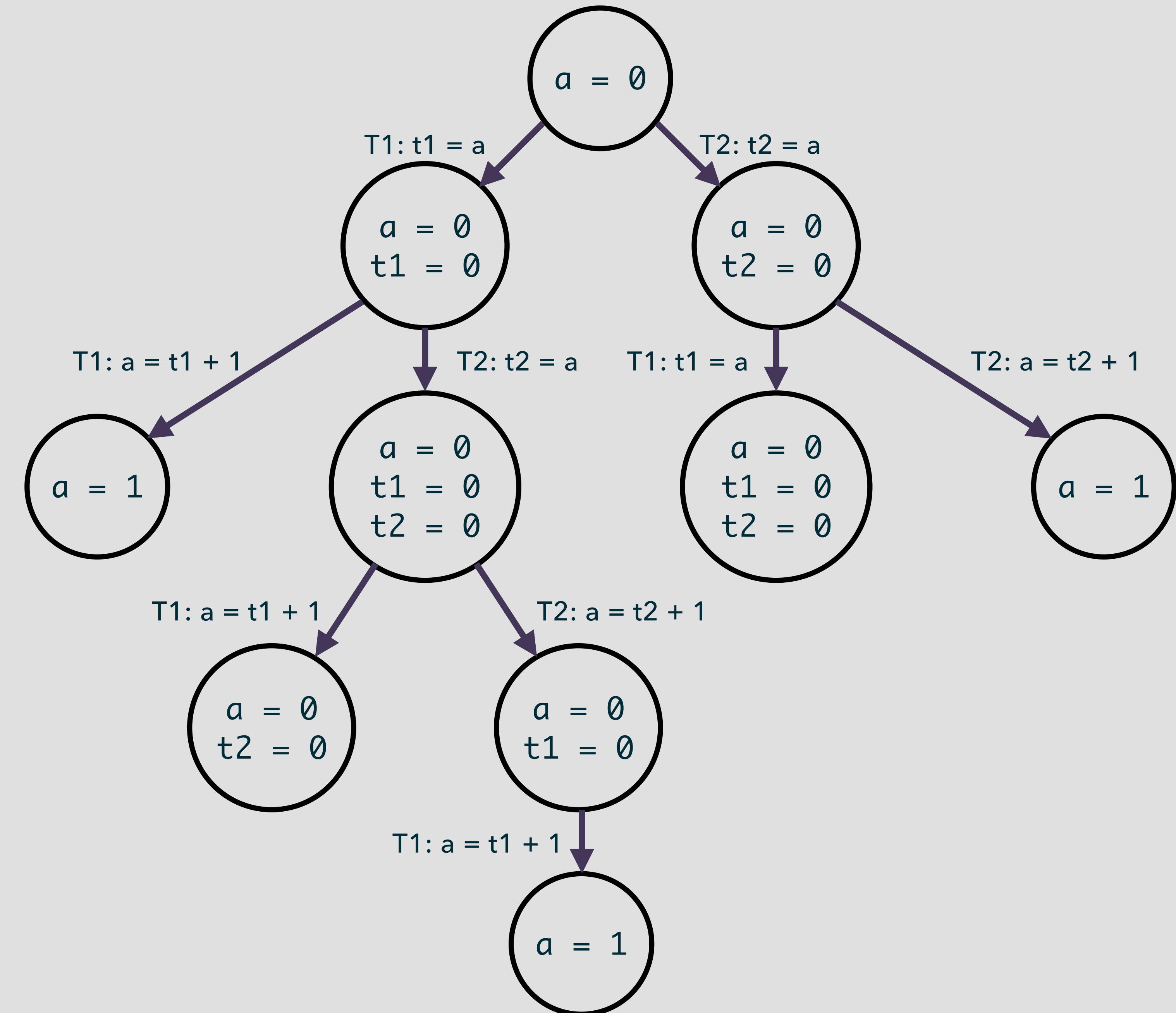
def T2():
    global a
    while True:
        t2 = a
        a = t2 + 1
```



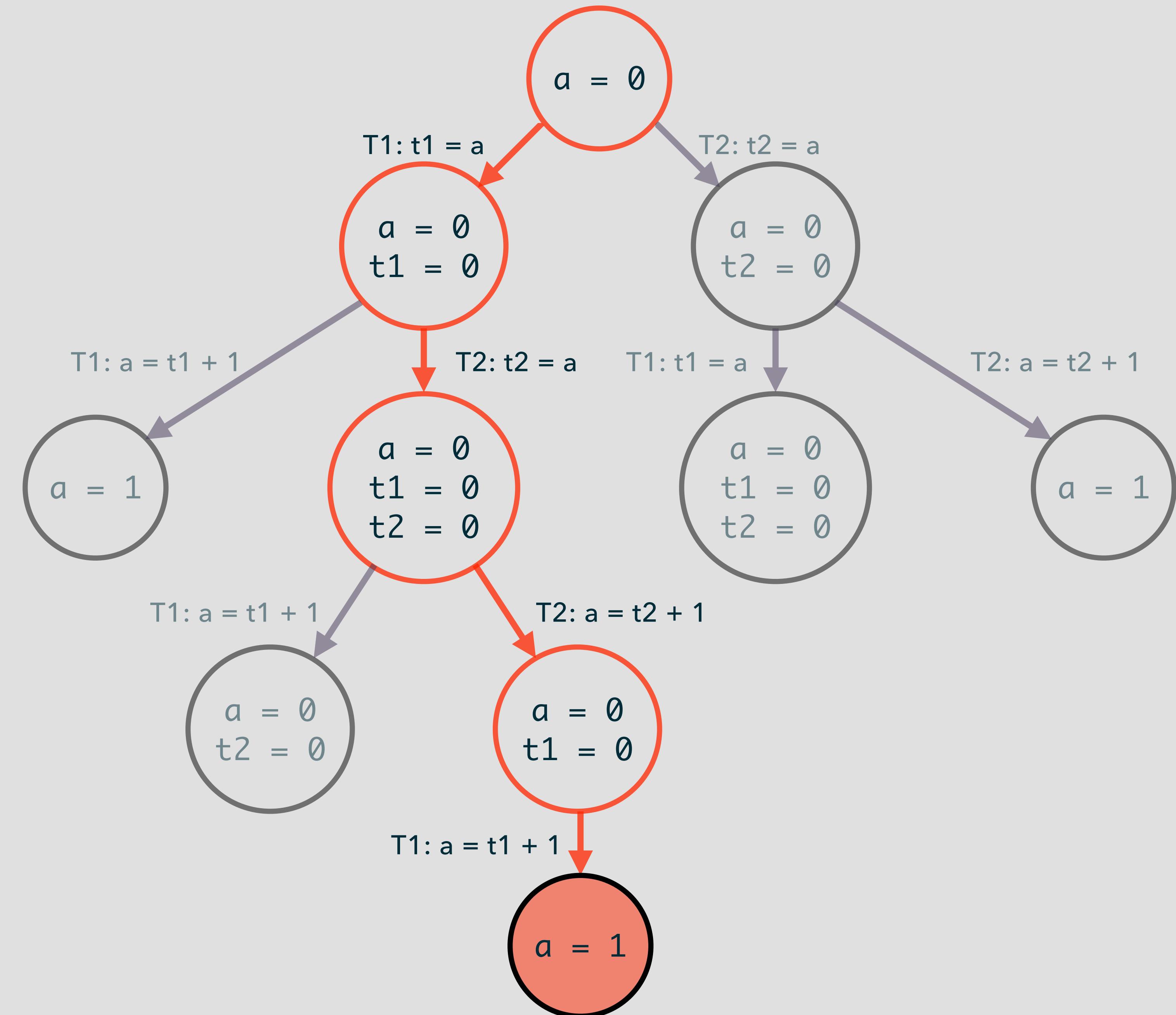
```
a = 0

def T1():
    global a
    while True:
        t1 = a
        a = t1 + 1

def T2():
    global a
    while True:
        t2 = a
        a = t2 + 1
```



```
a = 0  
  
def T1():  
    global a  
    while True:  
        t1 = a  
        a = t1 + 1  
  
def T2():  
    global a  
    while True:  
        t2 = a  
        a = t2 + 1
```



# Peterson algorithm

```
flag[0] = true;  
turn = 1;  
while (flag[1] == true && turn == 1) { wait(); }  
// Critical section  
flag[0] = false;
```

```
flag[1] = true;  
turn = 0;  
while (flag[0] == true && turn == 0) { wait(); }  
// Critical section  
flag[1] = false;
```

# Multithreading issue

- Naive “alipay” – data race
- Dining philosopher

Suting Chen