

# Function Calls & Stack examples

---

Stefan Mangard

November 10, 2021

Computer Organization and Networks  
Graz University of Technology

## Push and Pop

---



# Push and pop

```
ADDI x2, x0, 0x700
```

```
ADDI x5, x0, 1
```

```
ADDI x6, x0, 2
```

```
ADDI x7, x0, 3
```

```
ADDI x2, x2, -4
```

```
SW x5, 0(x2)
```

```
ADDI x2, x2, -4
```

```
SW x6, 0(x2)
```

```
ADDI x2, x2, -4
```

```
SW x7, 0(x2)
```

```
LW x7, 0(x2)
```

```
ADDI x2, x2, 4
```

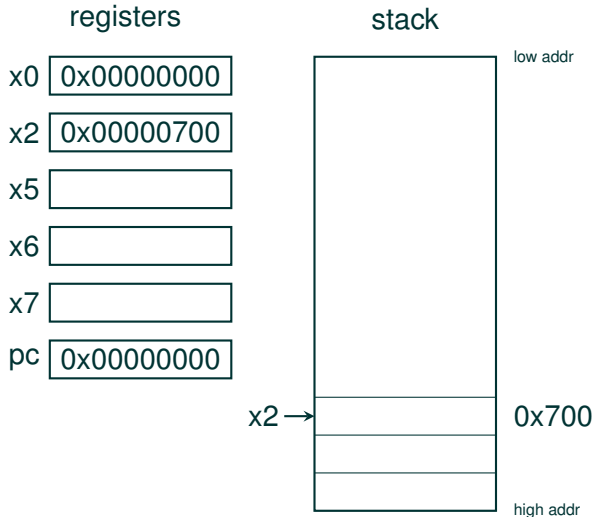
```
LW x6, 0(x2)
```

```
ADDI x2, x2, 4
```

```
LW x5, 0(x2)
```

```
ADDI x2, x2, 4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

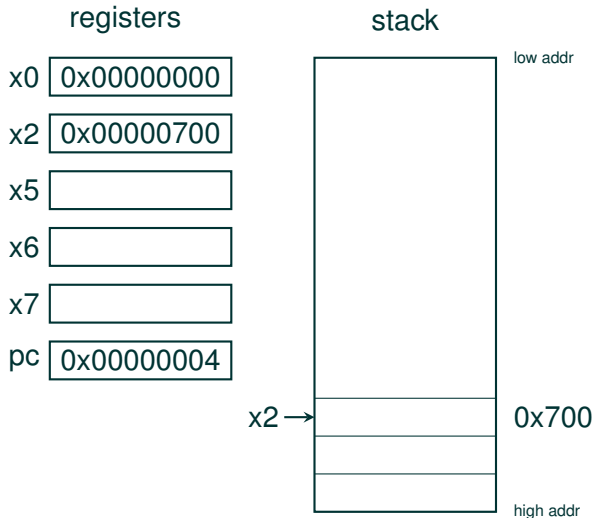
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

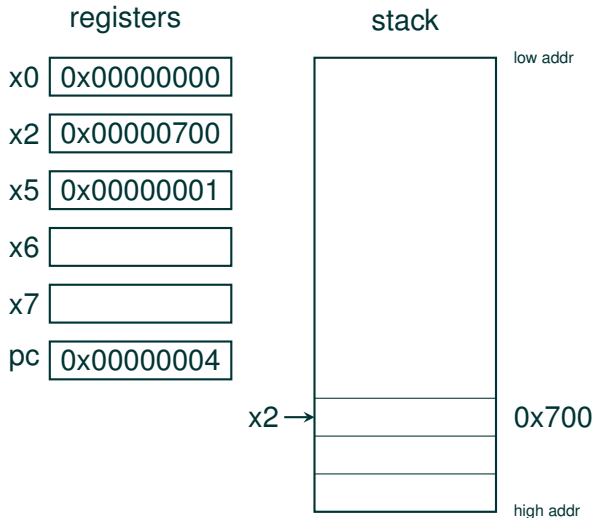
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

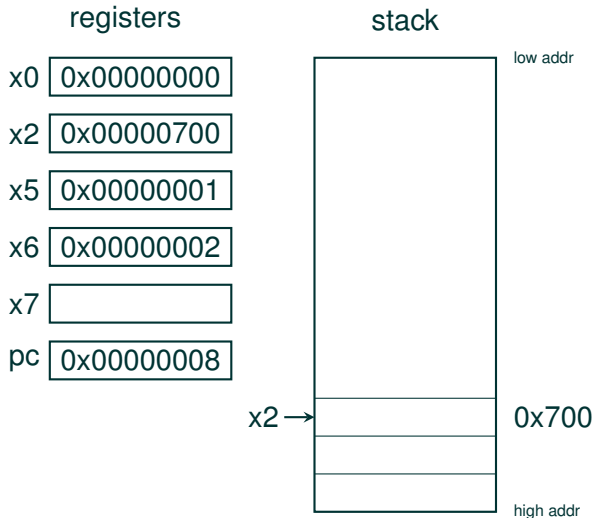
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

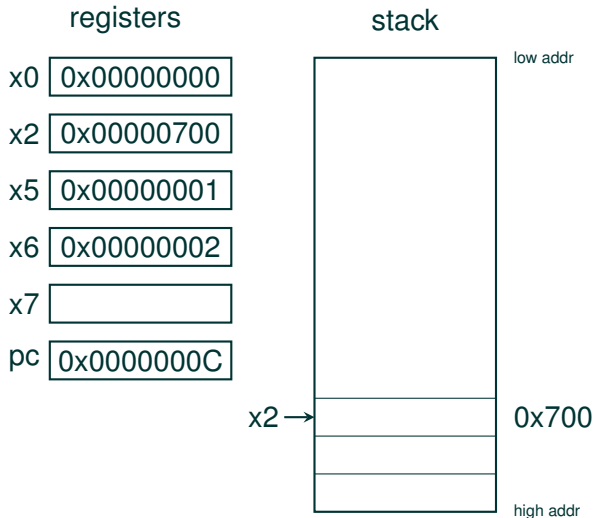
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

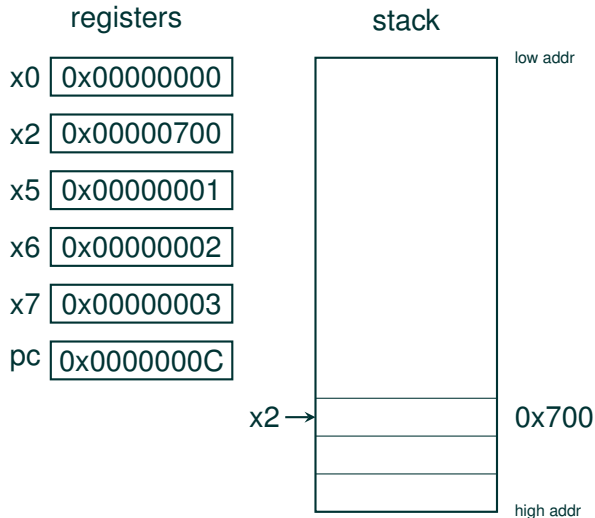
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

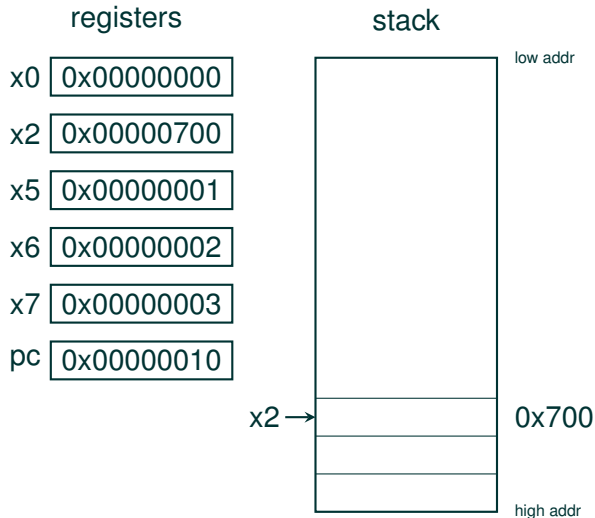
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

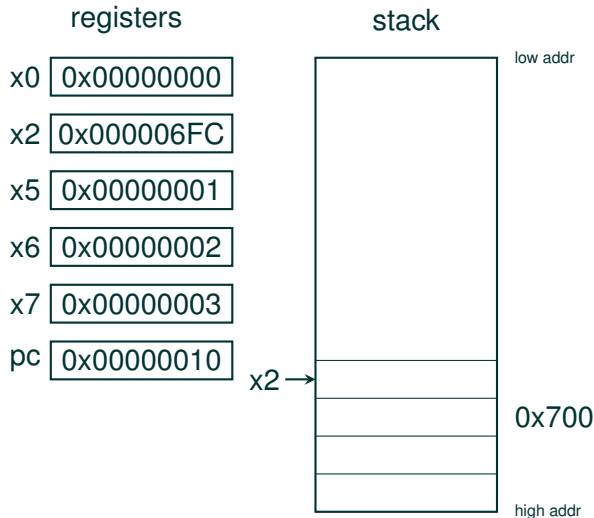
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

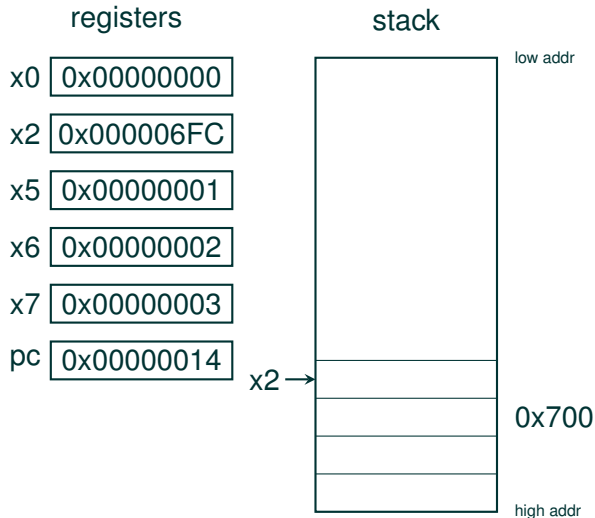
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

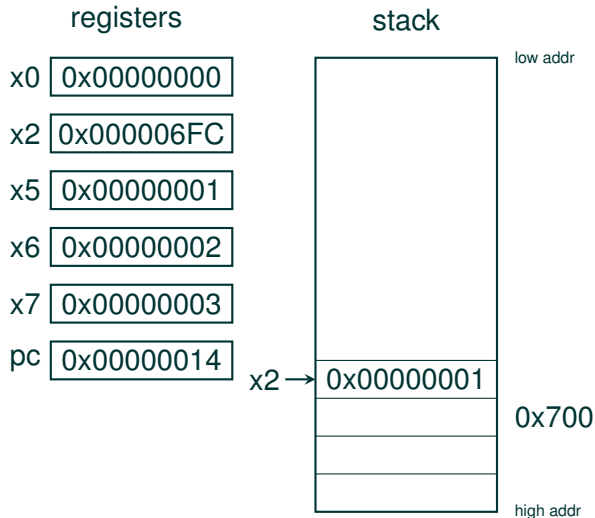
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

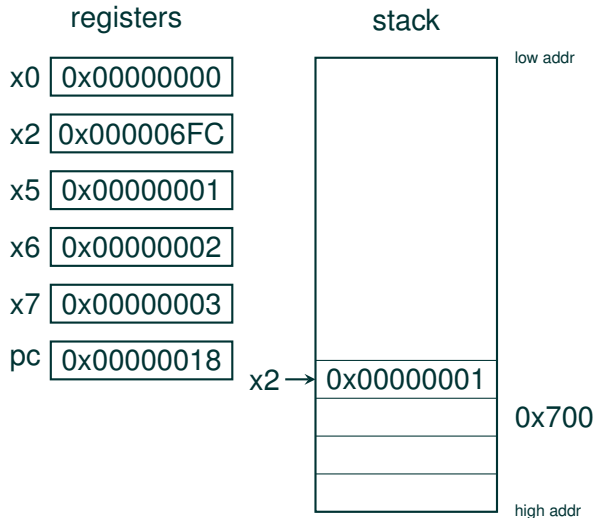
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

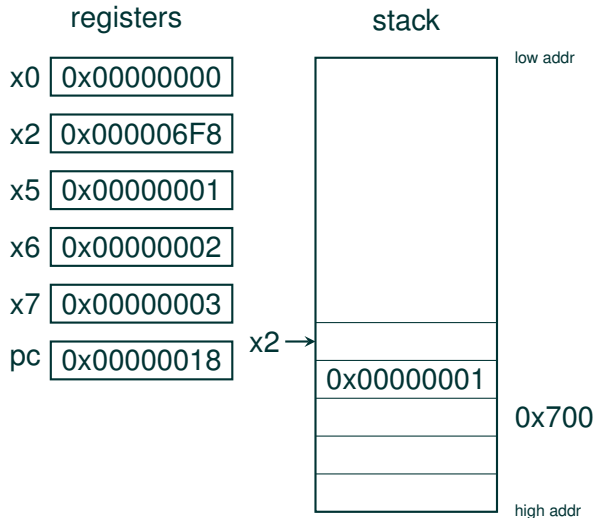
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

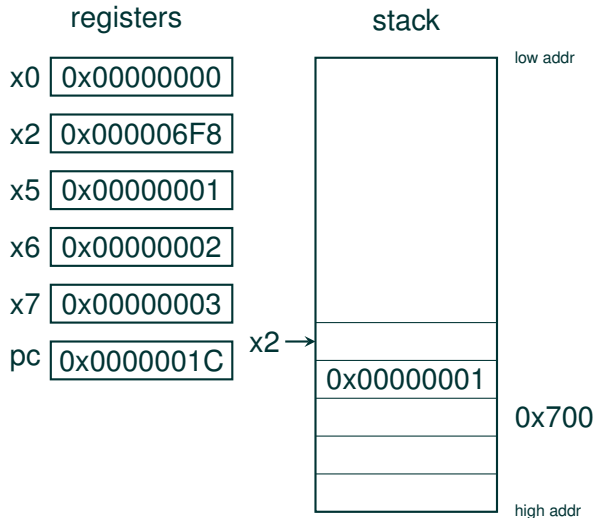
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```







# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

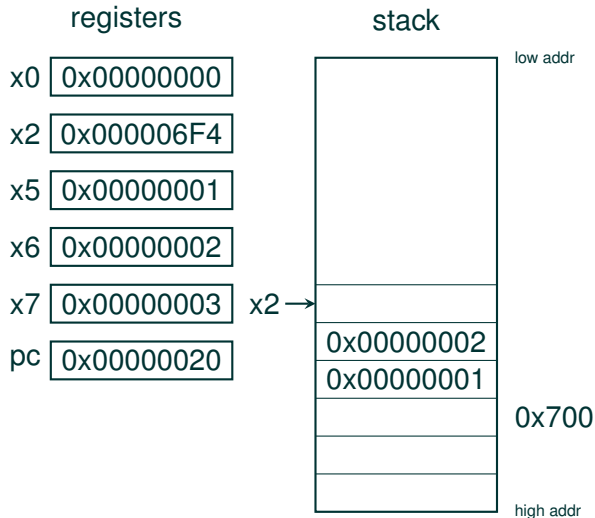
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```







# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

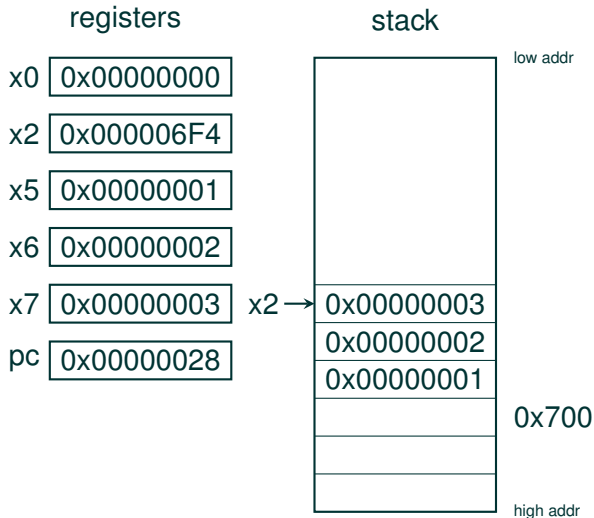
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

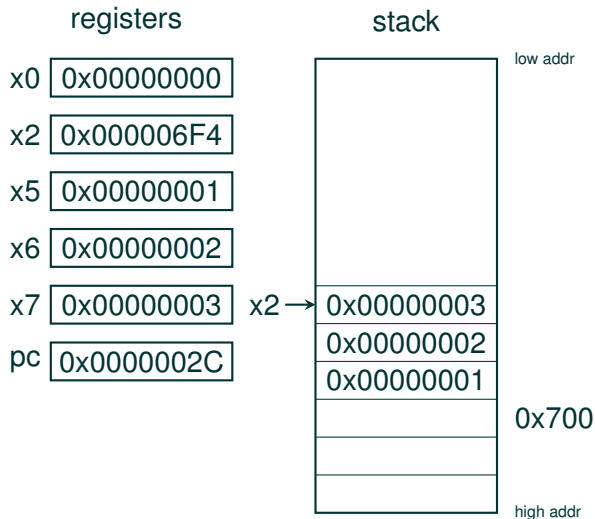
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

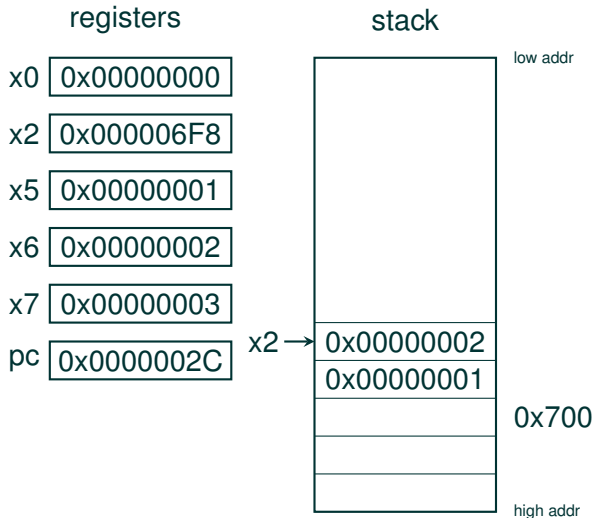
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

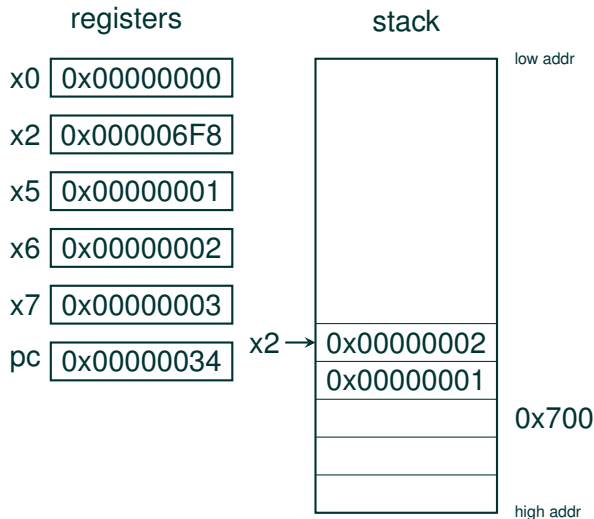
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

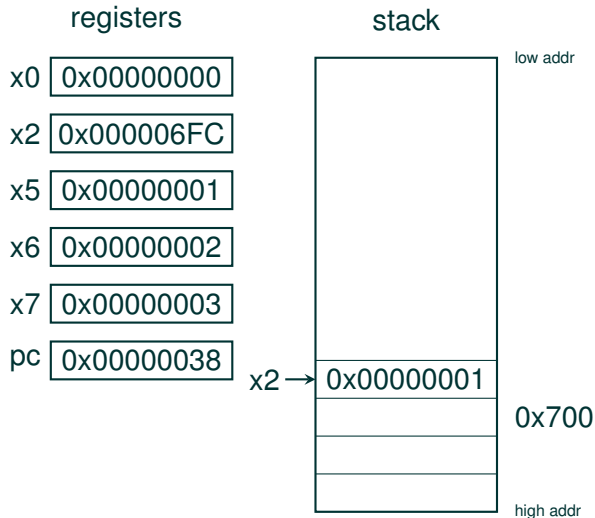
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,-(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

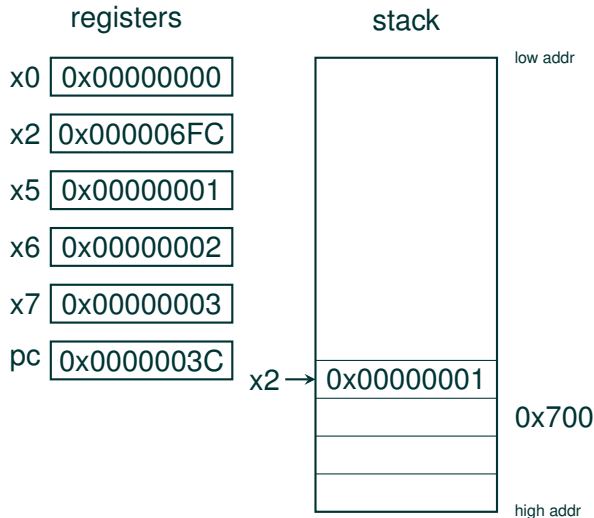
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```

registers

x0 0x00000000

x2 0x00000700

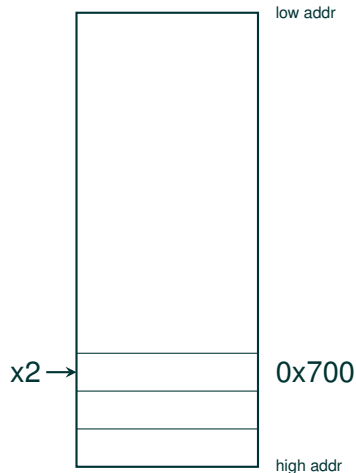
x5 0x00000001

x6 0x00000002

x7 0x00000003

pc 0x00000040

stack





# Push and pop

```
ADDI x2,x0,0x700
```

```
ADDI x5,x0,1
```

```
ADDI x6,x0,2
```

```
ADDI x7,x0,3
```

```
ADDI x2,x2,-4
```

```
SW x5,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x6,0(x2)
```

```
ADDI x2,x2,-4
```

```
SW x7,0(x2)
```

```
LW x7,0(x2)
```

```
ADDI x2,x2,4
```

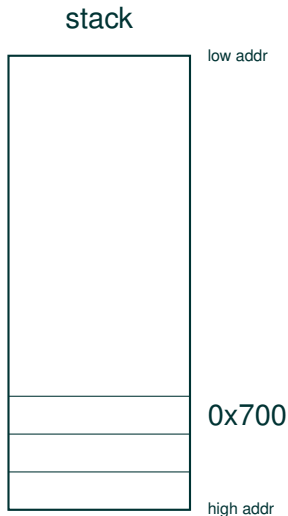
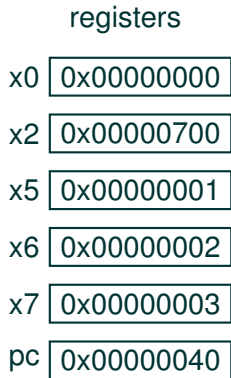
```
LW x6,0(x2)
```

```
ADDI x2,x2,4
```

```
LW x5,0(x2)
```

```
ADDI x2,x2,4
```

```
EBREAK
```



## **Recursive call on a stack**

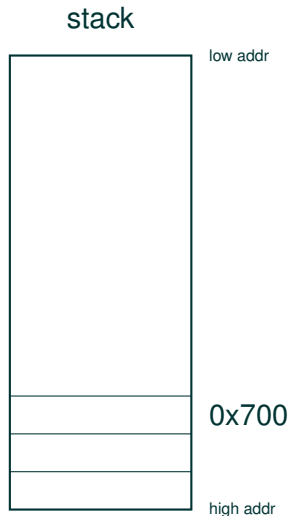
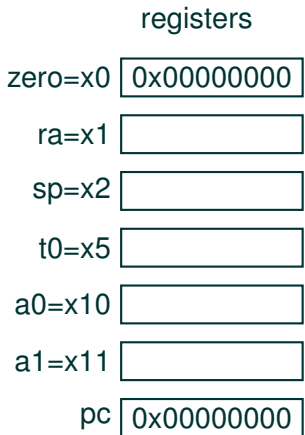
---

## Recursive call on a stack

```
// Computes the sum of the arithmetic progression defined by  
// result = sum init+(n-1), n=[1,count]  
int arith_series(int init, int count) {  
    if (1 >= count)  
        return init;  
    return count + arith_series(init, count-1);  
}  
  
int main(void) {  
    return arith_series(1, 3);  
}
```

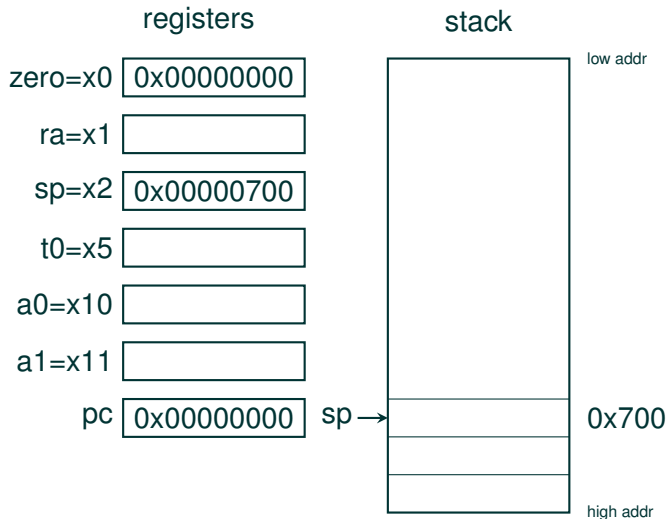
# Recursive call on a stack

```
_start:  
  ADDI sp, zero, 0x700  
  JAL ra, main  
  EBREAK  
arith_series:  
  ADDI sp, sp, -8  
  SW ra, 4(sp)  
  ADDI t0, zero, 1  
  BGE t0, a1, arith_series_return  
  SW a1, 0(sp)  
  ADDI a1, a1, -1  
  JAL ra, arith_series  
  LW a1, 0(sp)  
  ADD a0, a0, a1  
arith_series_return:  
  LW ra, 4(sp)  
  ADDI sp, sp, 8  
  JALR zero, 0(ra)  
main:  
  ADDI sp, sp, -4  
  SW ra, 0(sp)  
  ADDI a1, zero, 3  
  ADDI a0, zero, 1  
  JAL ra, arith_series  
  LW ra, 0(sp)  
  ADDI sp, sp, 4  
  JALR zero, 0(ra)
```



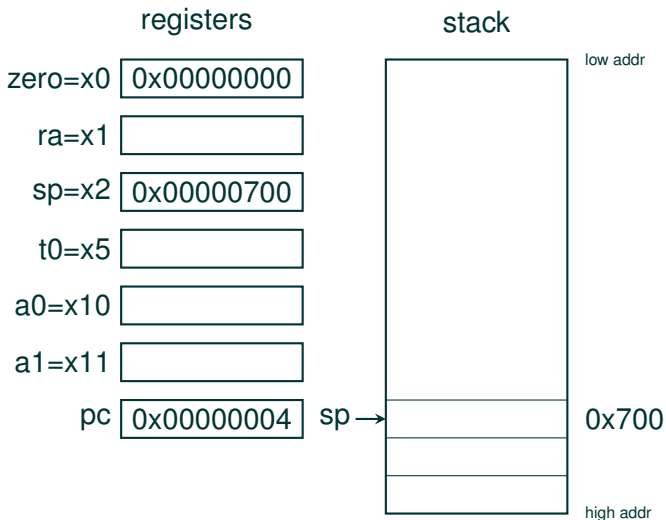
# Recursive call on a stack

```
_start:  
  ADDI sp, zero, 0x700  
  JAL ra, main  
  EBREAK  
arith_series:  
  ADDI sp, sp, -8  
  SW ra, 4(sp)  
  ADDI t0, zero, 1  
  BGE t0, a1, arith_series_return  
  SW a1, 0(sp)  
  ADDI a1, a1, -1  
  JAL ra, arith_series  
  LW a1, 0(sp)  
  ADD a0, a0, a1  
arith_series_return:  
  LW ra, 4(sp)  
  ADDI sp, sp, 8  
  JALR zero, 0(ra)  
main:  
  ADDI sp, sp, -4  
  SW ra, 0(sp)  
  ADDI a1, zero, 3  
  ADDI a0, zero, 1  
  JAL ra, arith_series  
  LW ra, 0(sp)  
  ADDI sp, sp, 4  
  JALR zero, 0(ra)
```



# Recursive call on a stack

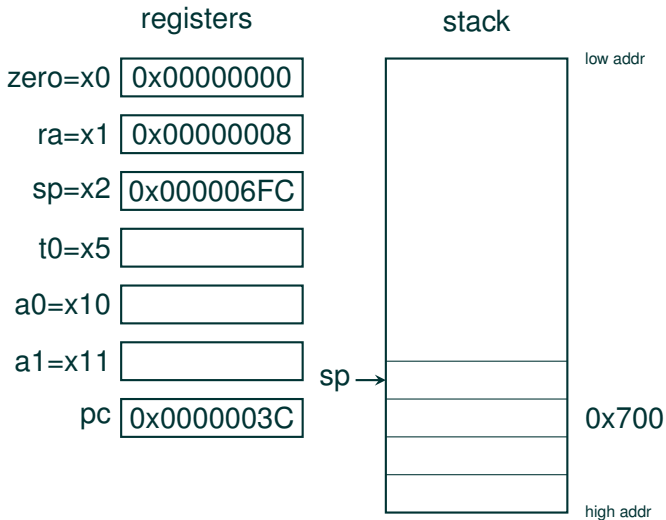
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





# Recursive call on a stack

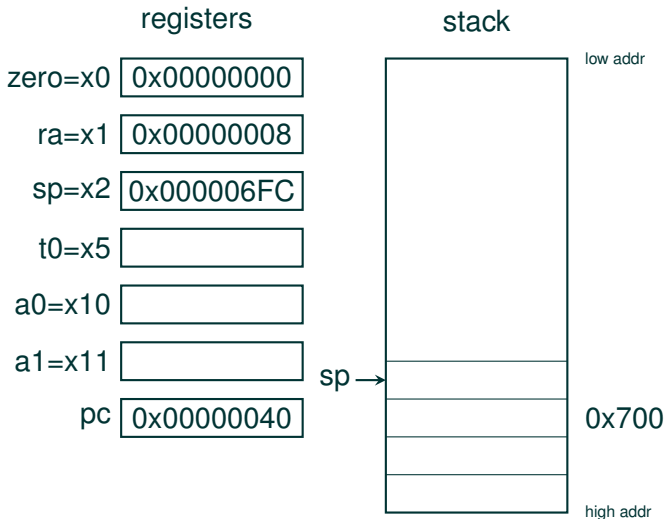
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```

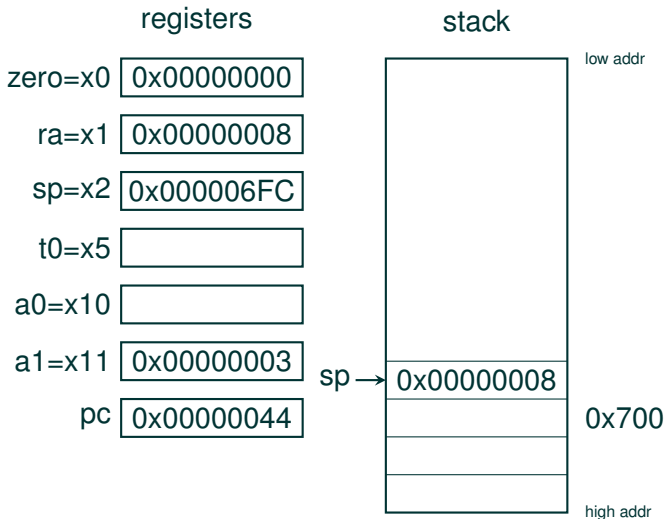






# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



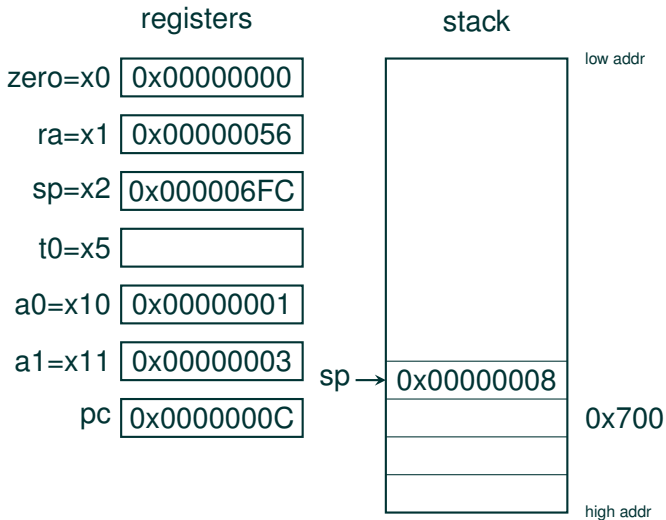






# Recursive call on a stack

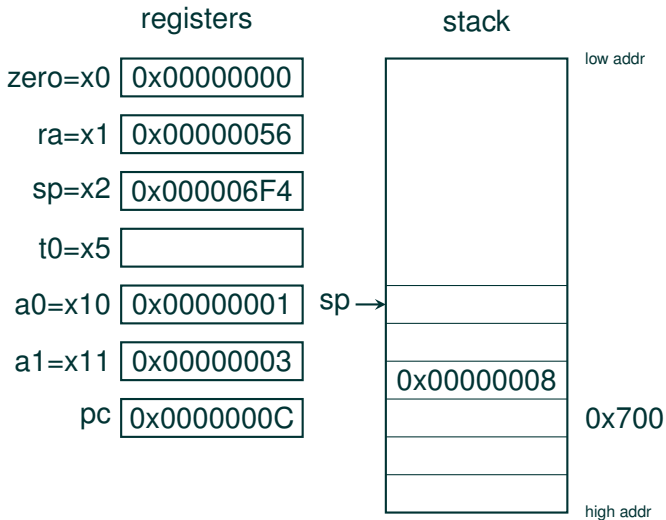
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





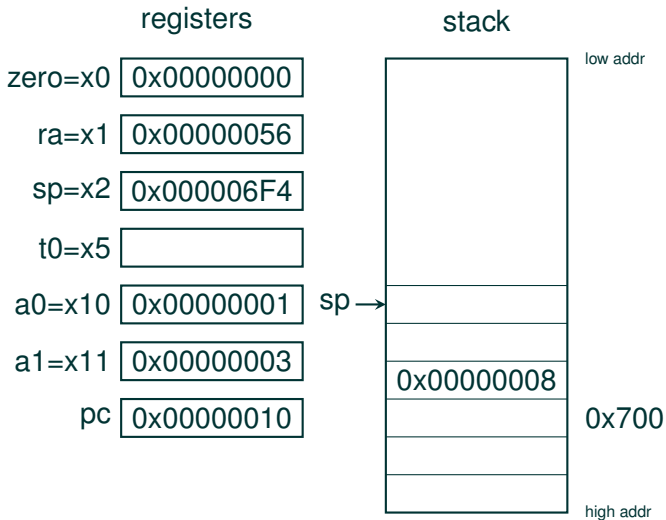
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

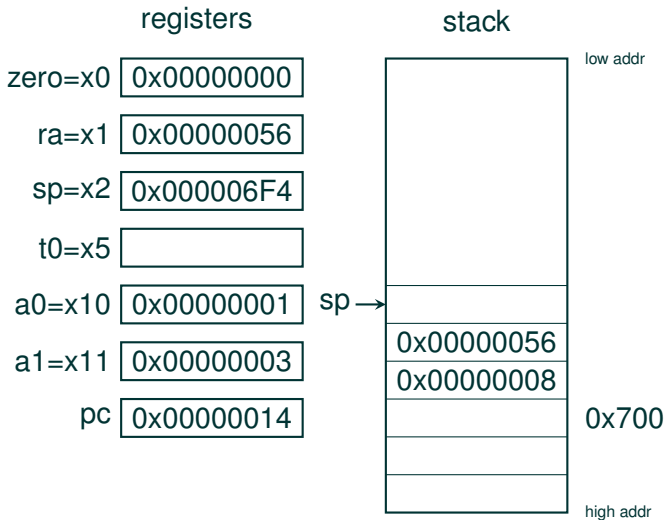
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 0(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





# Recursive call on a stack

```
_start:  
  ADDI sp, zero, 0x700  
  JAL ra, main  
  EBREAK  
arith_series:  
  ADDI sp, sp, -8  
  SW ra, 4(sp)  
  ADDI t0, zero, 1  
  BGE t0, a1, arith_series_return  
  SW a1, 0(sp)  
  ADDI a1, a1, -1  
  JAL ra, arith_series  
  LW a1, 0(sp)  
  ADD a0, a0, a1  
arith_series_return:  
  LW ra, 4(sp)  
  ADDI sp, sp, 8  
  JALR zero, 0(ra)  
main:  
  ADDI sp, sp, -4  
  SW ra, 0(sp)  
  ADDI a1, zero, 3  
  ADDI a0, zero, 1  
  JAL ra, arith_series  
  LW ra, 0(sp)  
  ADDI sp, sp, 4  
  JALR zero, 0(ra)
```









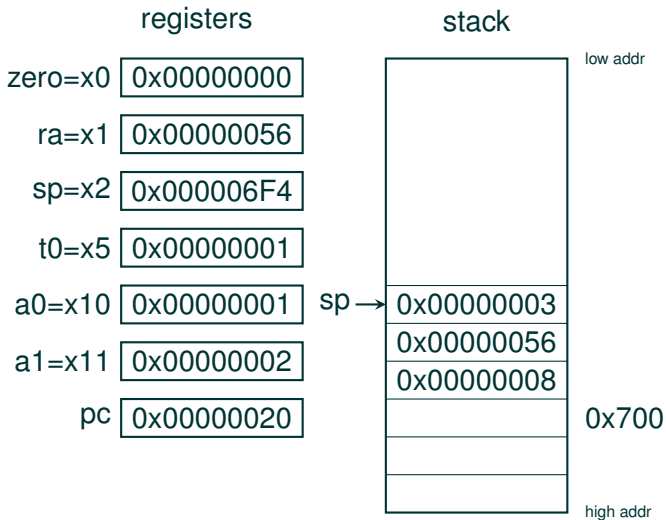






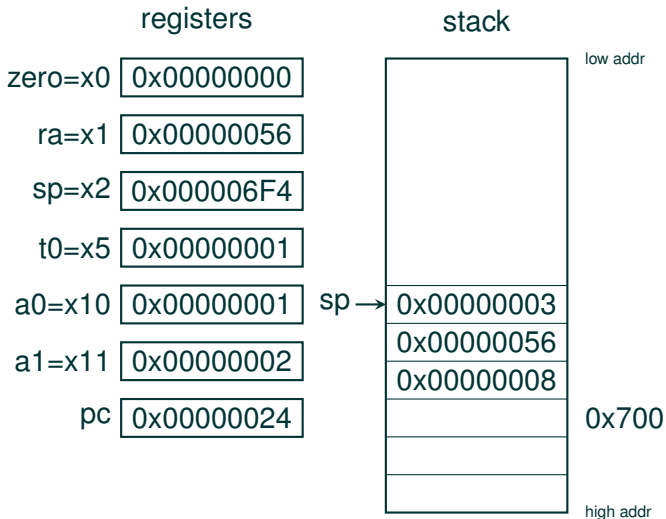
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



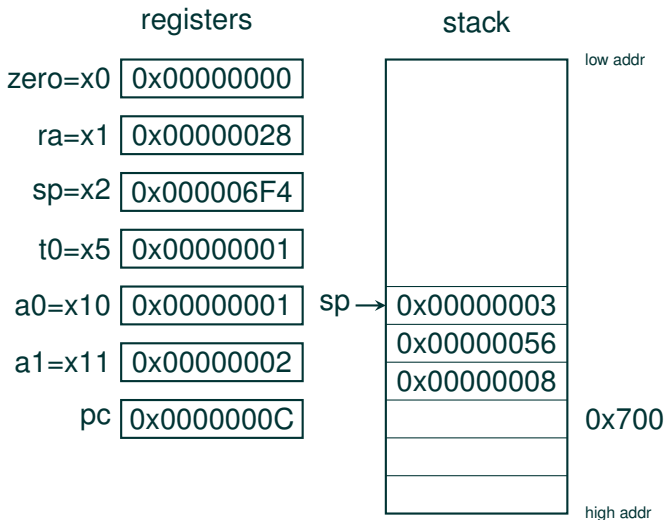
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



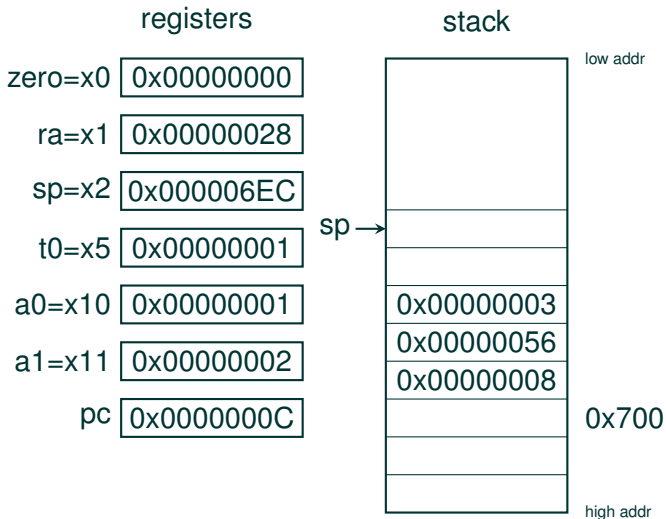
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



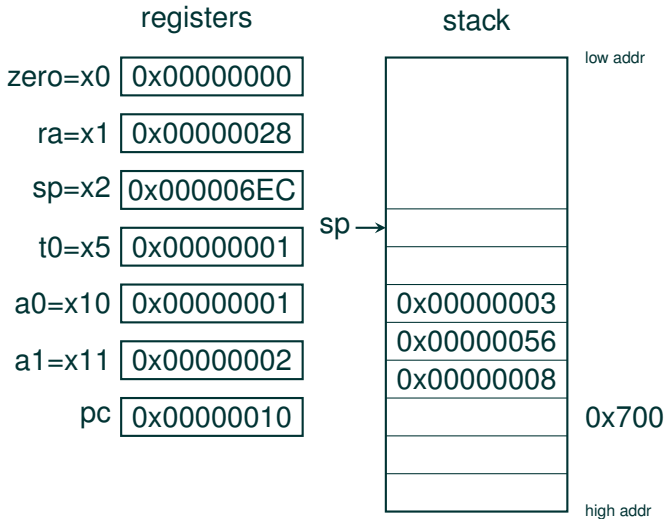
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



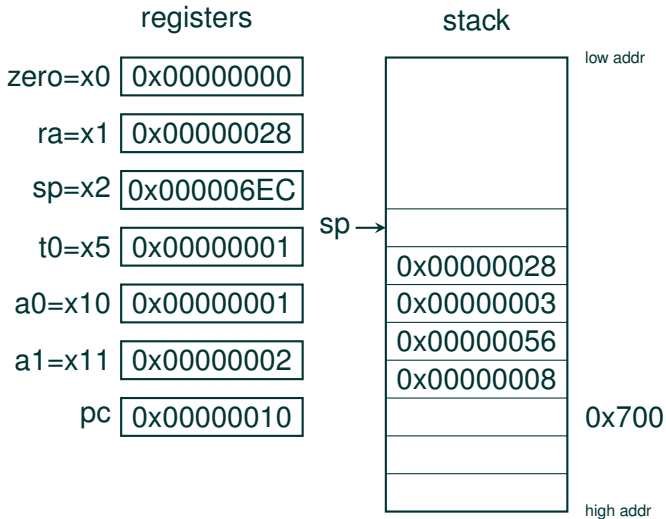
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 0(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



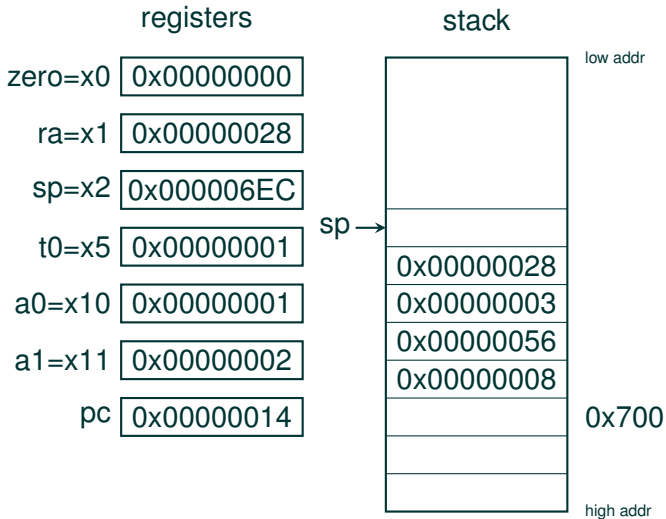
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 0(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```

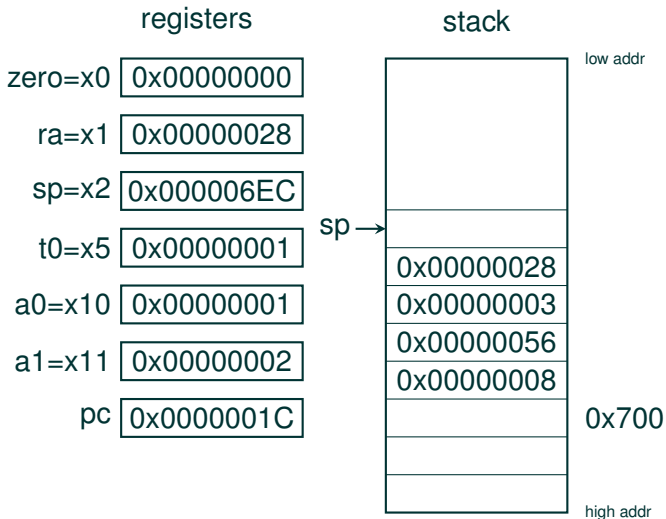






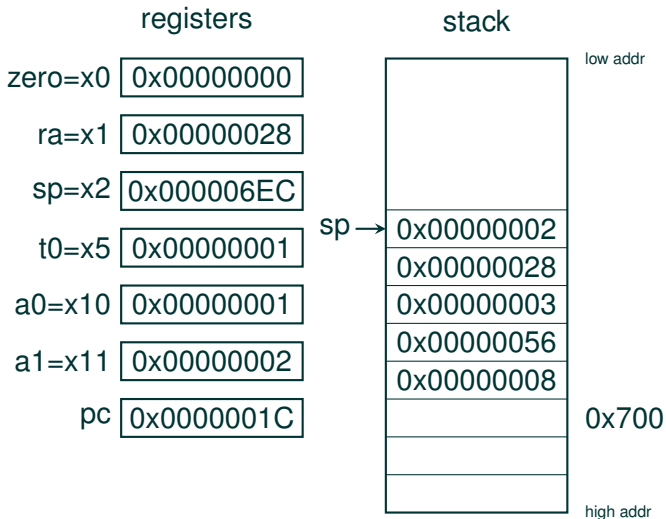
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



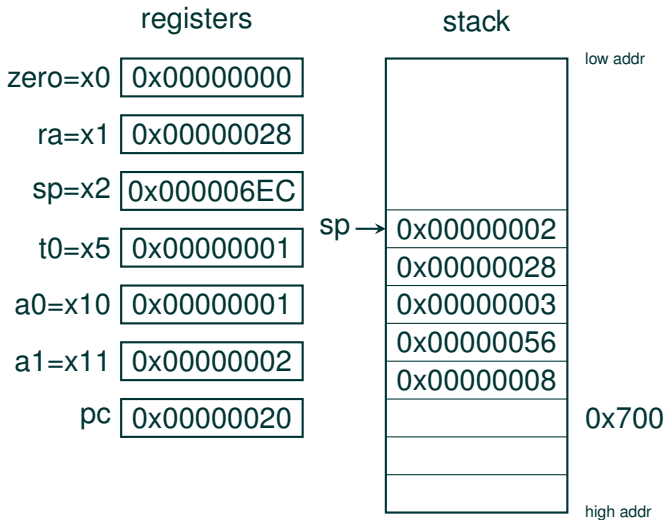
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



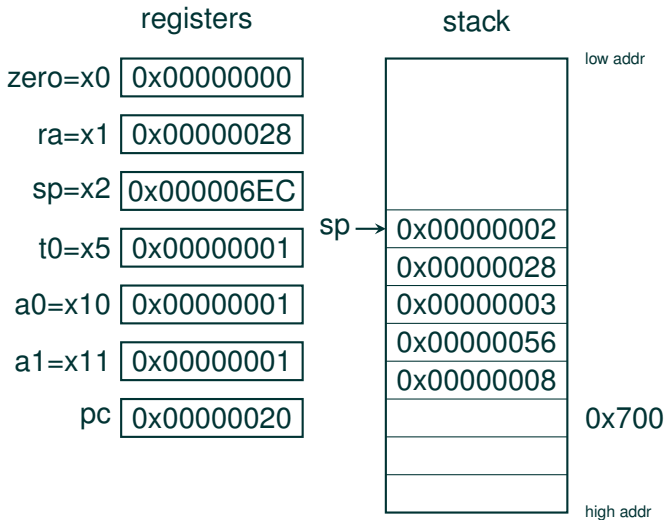
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



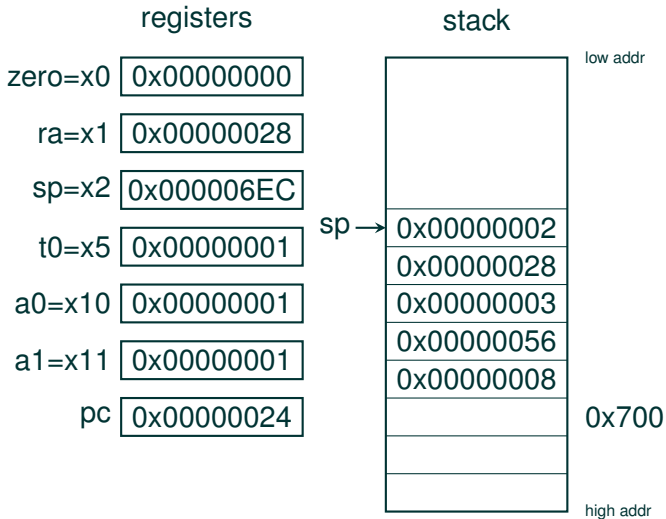
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



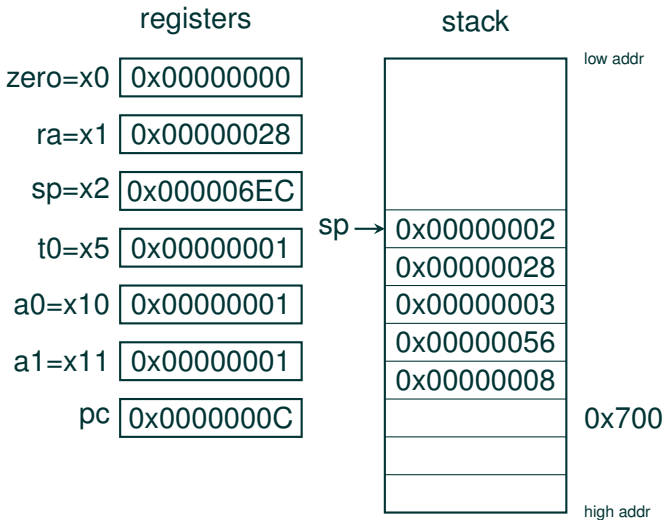
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



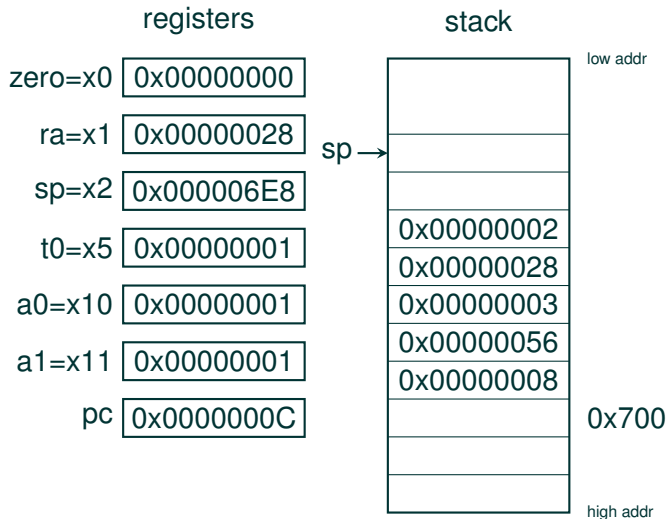
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

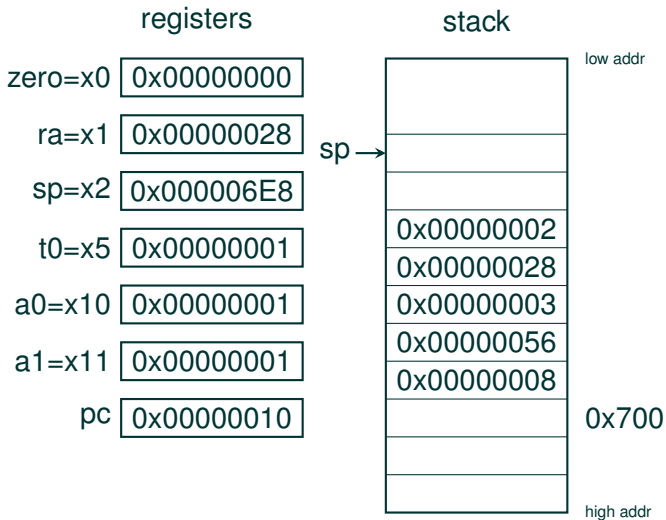
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





# Recursive call on a stack

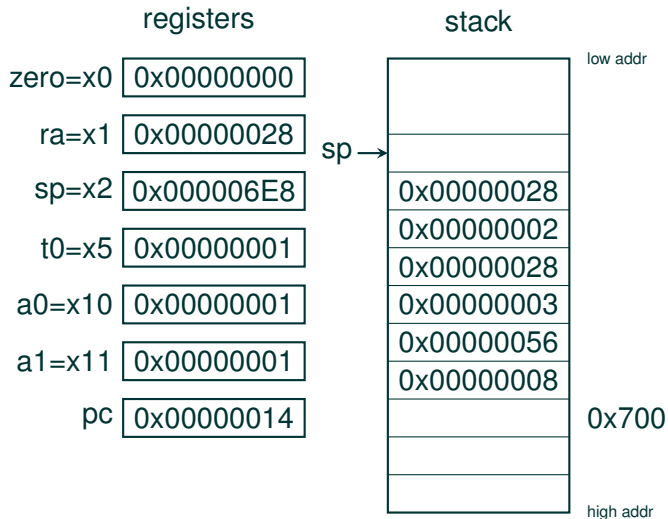
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 0(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





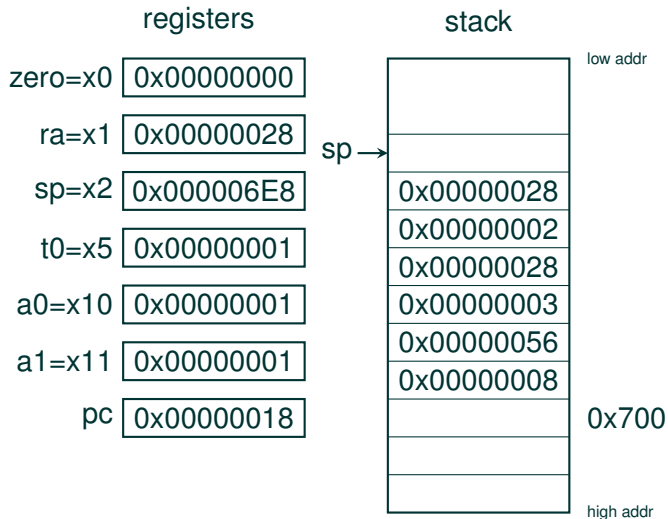
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



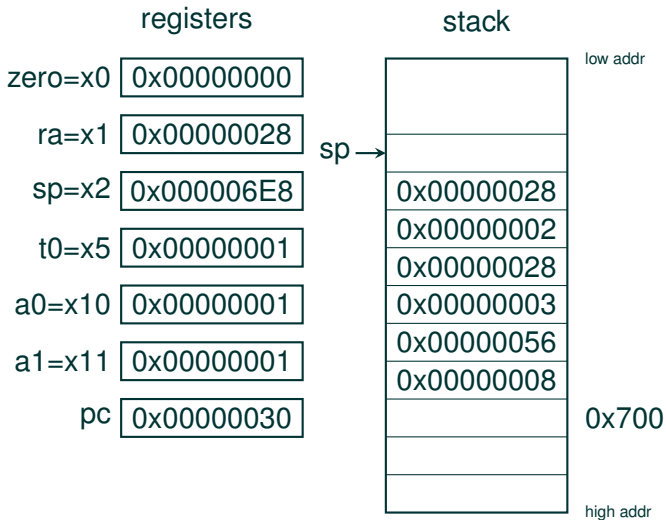
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

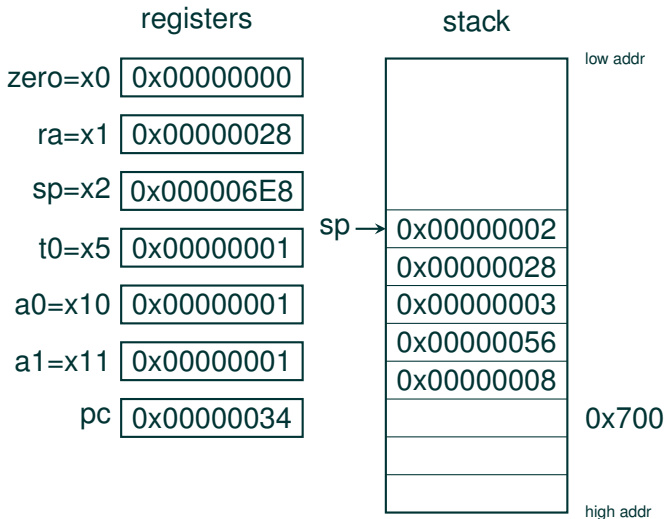
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





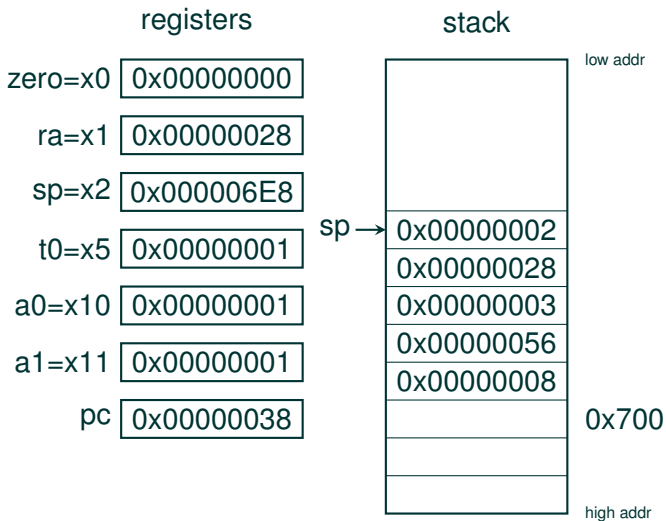
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

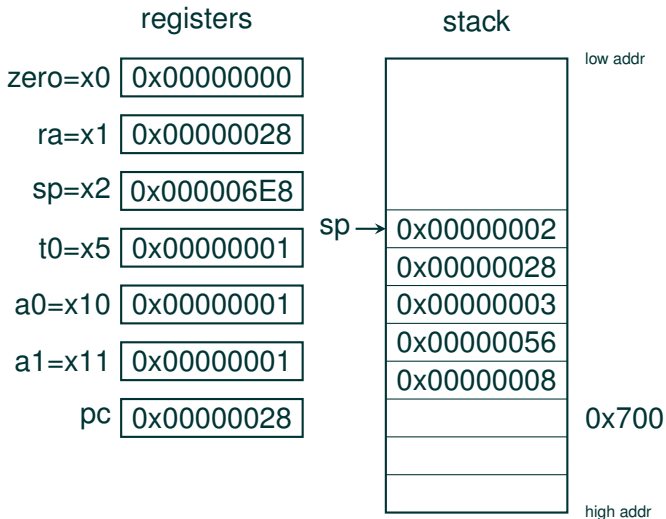
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





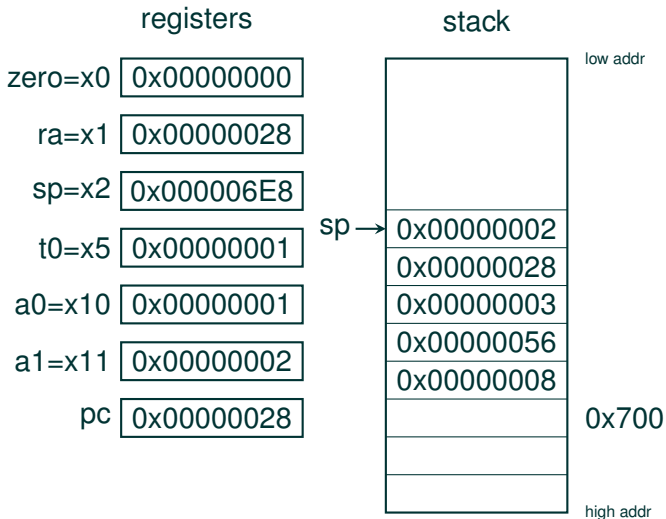
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



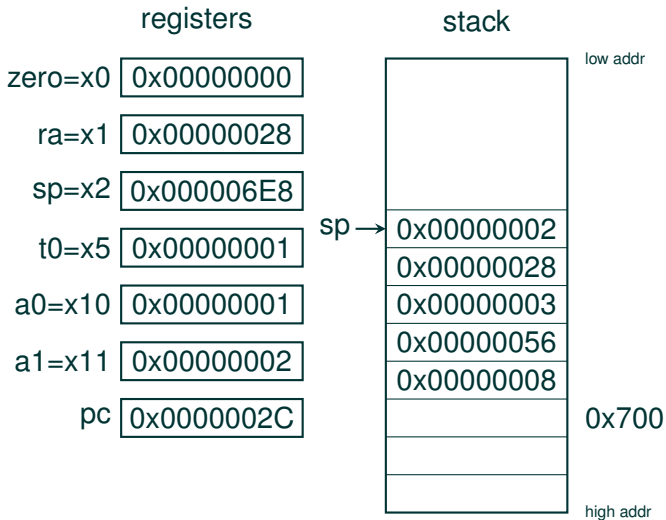
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



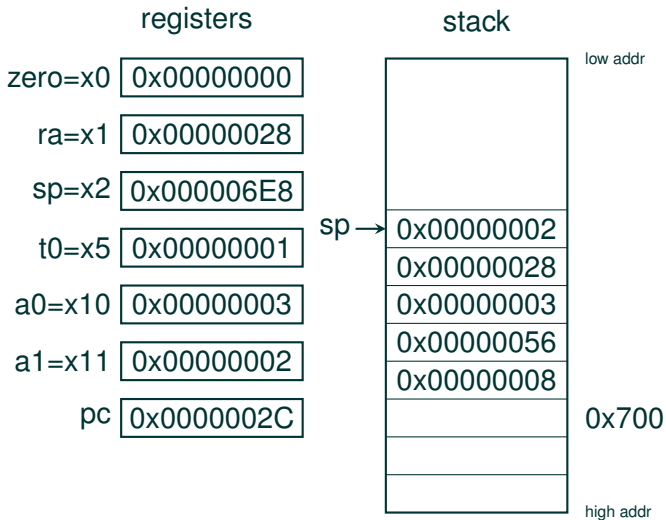
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



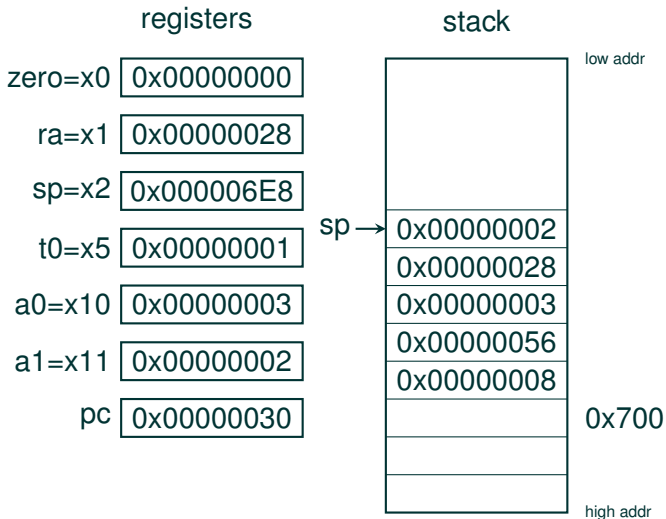
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



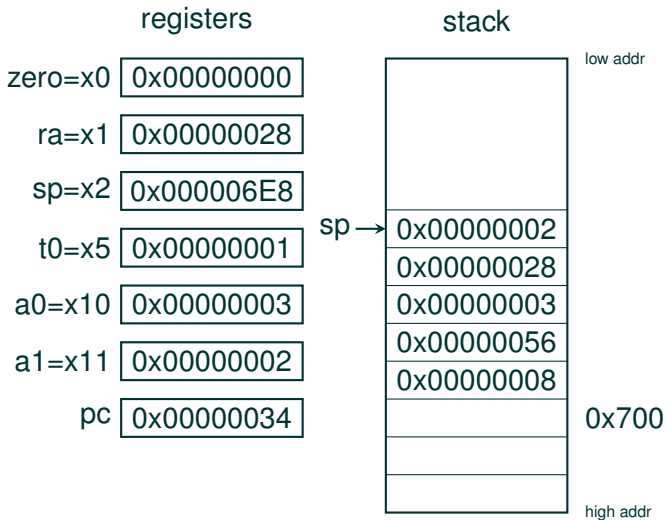
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



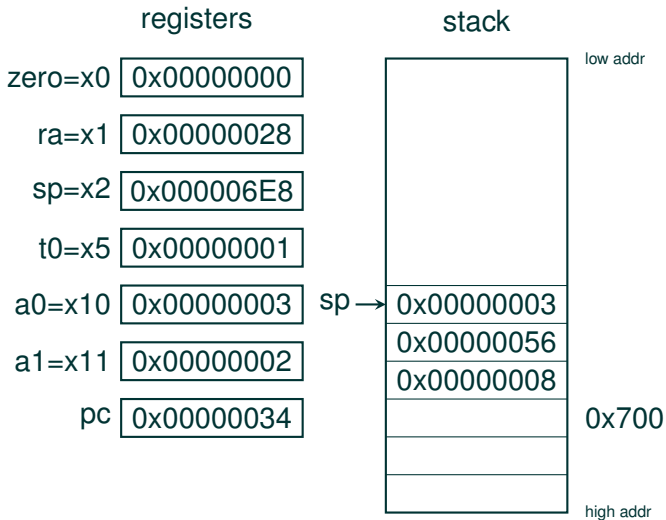
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



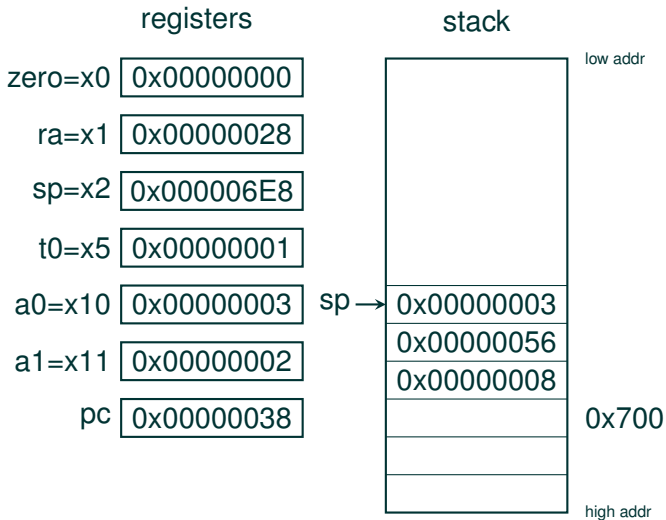
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

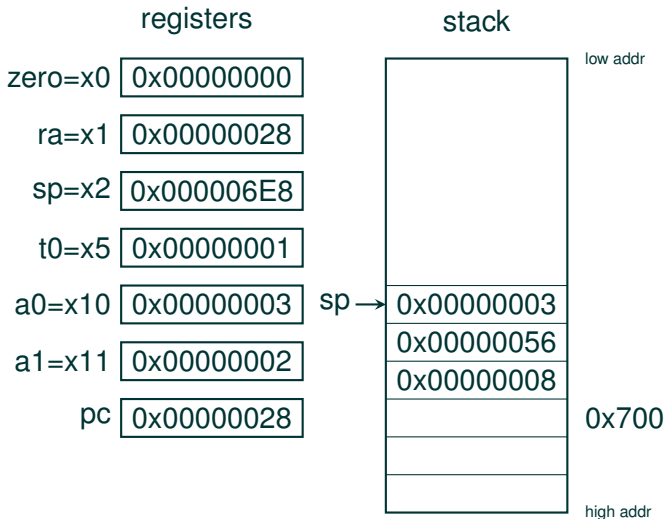
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





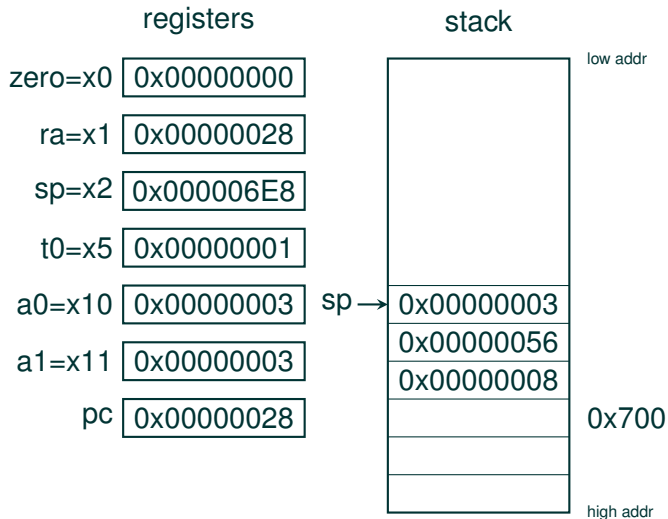
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



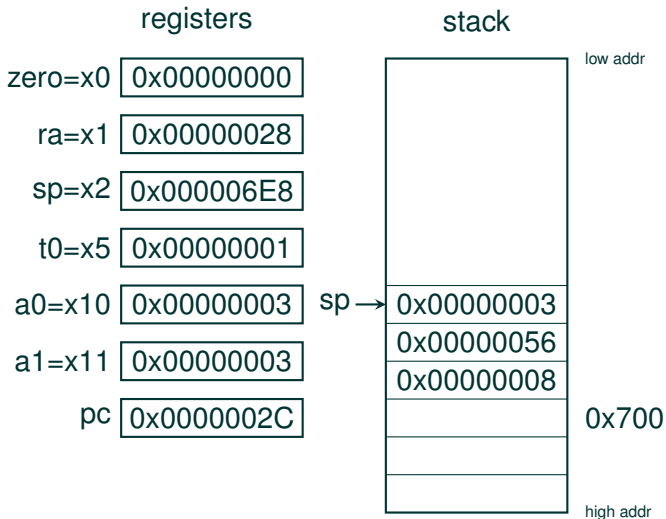
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



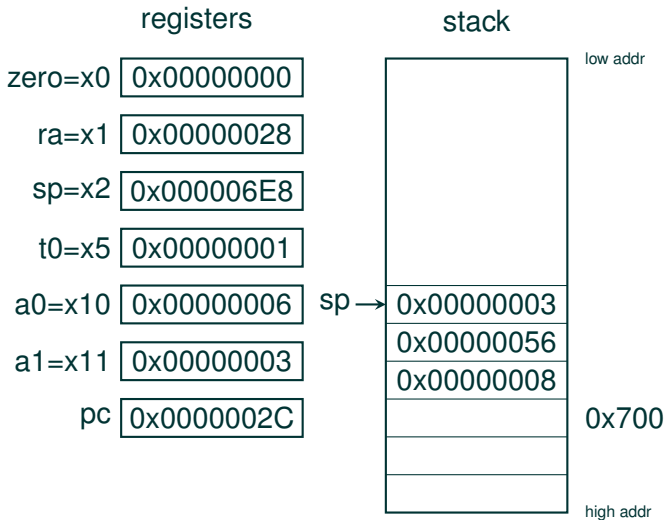
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



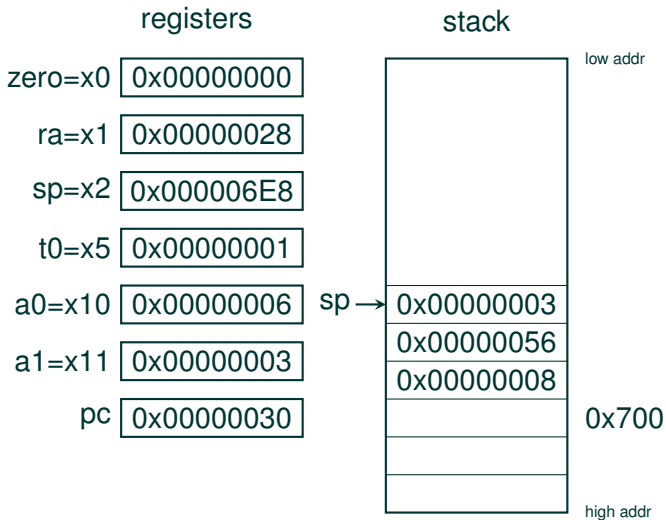
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



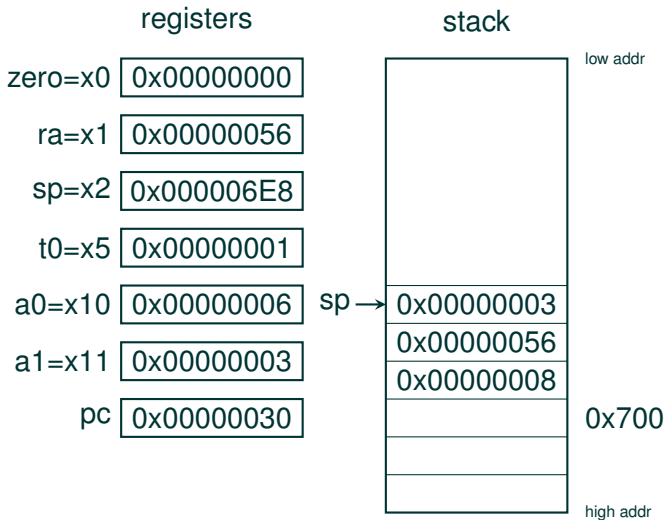
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

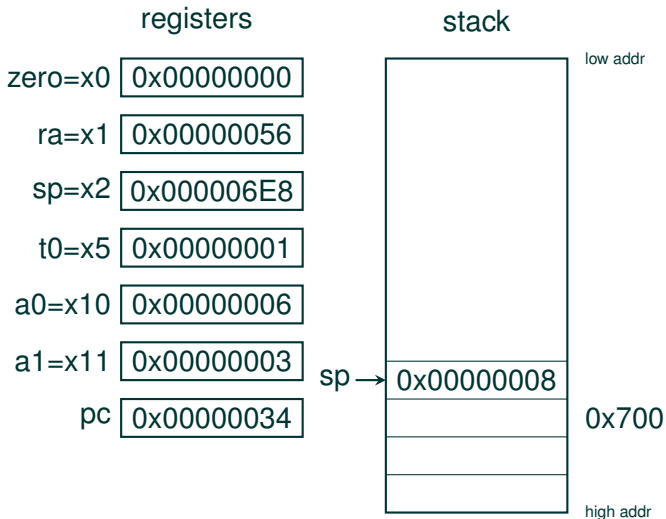
```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```





# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```







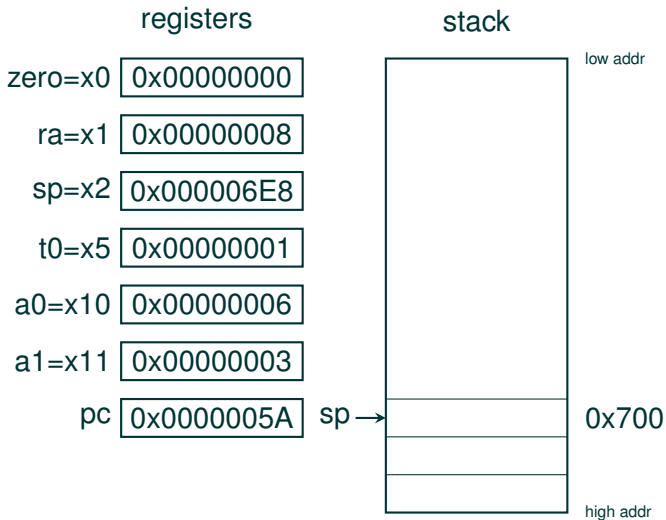






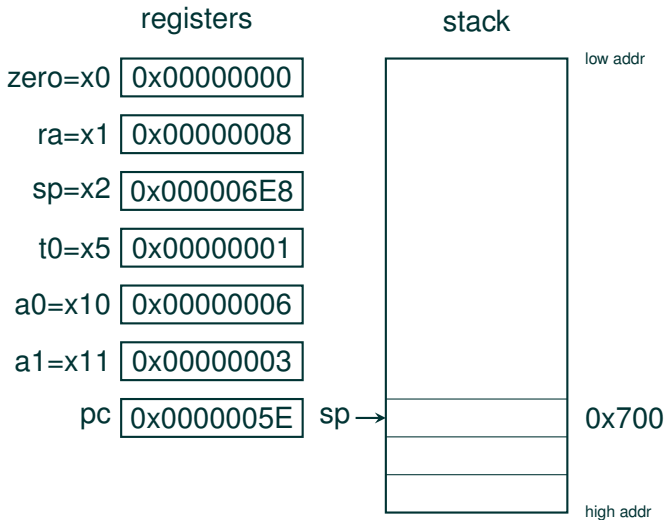
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



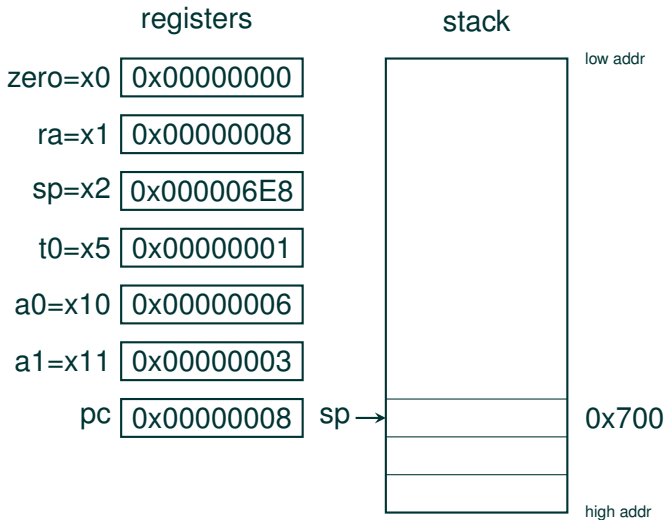
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



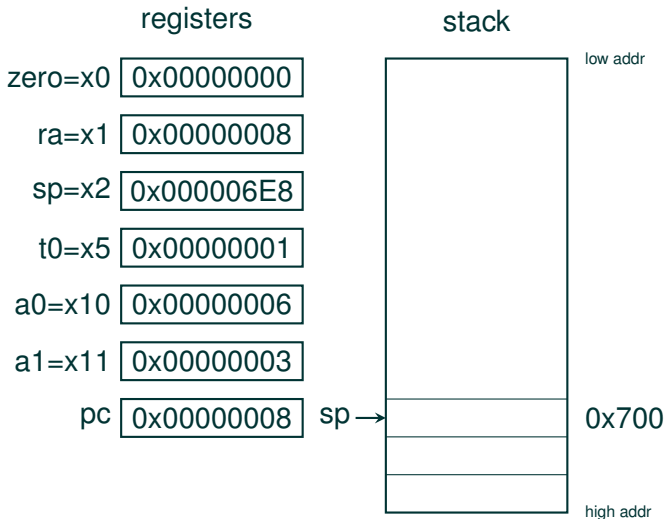
# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



# Recursive call on a stack

```
_start:
    ADDI sp, zero, 0x700
    JAL ra, main
    EBREAK
arith_series:
    ADDI sp, sp, -8
    SW ra, 4(sp)
    ADDI t0, zero, 1
    BGE t0, a1, arith_series_return
    SW a1, 0(sp)
    ADDI a1, a1, -1
    JAL ra, arith_series
    LW a1, 0(sp)
    ADD a0, a0, a1
arith_series_return:
    LW ra, 4(sp)
    ADDI sp, sp, 8
    JALR zero, 0(ra)
main:
    ADDI sp, sp, -4
    SW ra, 0(sp)
    ADDI a1, zero, 3
    ADDI a0, zero, 1
    JAL ra, arith_series
    LW ra, 0(sp)
    ADDI sp, sp, 4
    JALR zero, 0(ra)
```



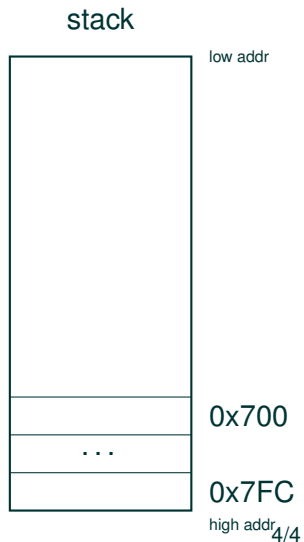
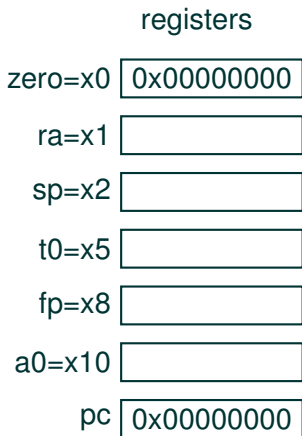


## **Buffer overflow**

---

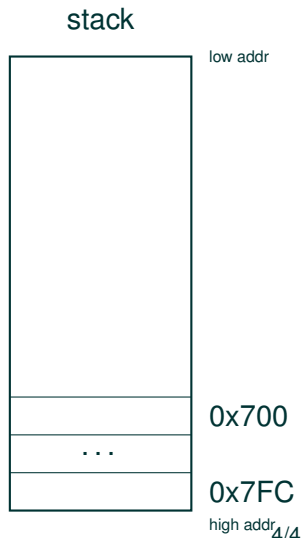
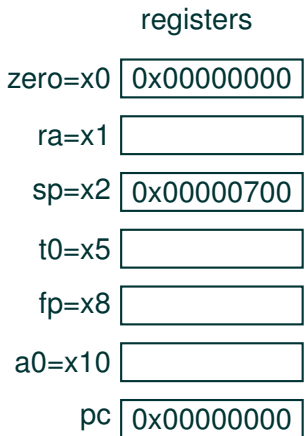
# Buffer overflow

```
_start:  
ADDI sp,zero,0x700  
ADDI fp,zero,0x700  
JAL ra,main  
EBREAK  
main:  
ADDI sp,sp,-8  
SW ra,4(sp)  
SW fp,0(sp)  
ADDI fp,sp,8  
JAL ra,vuln  
LW fp,0(sp)  
LW ra,4(sp)  
ADDI sp,sp,8  
JALR zero,0(ra)
```



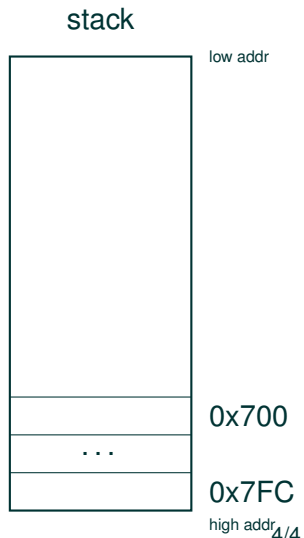
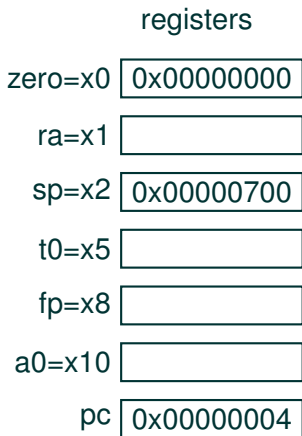
# Buffer overflow

```
_start:  
ADDI sp,zero,0x700  
ADDI fp,zero,0x700  
JAL ra,main  
EBREAK  
main:  
ADDI sp,sp,-8  
SW ra,4(sp)  
SW fp,0(sp)  
ADDI fp,sp,8  
JAL ra,vuln  
LW fp,0(sp)  
LW ra,4(sp)  
ADDI sp,sp,8  
JALR zero,0(ra)
```



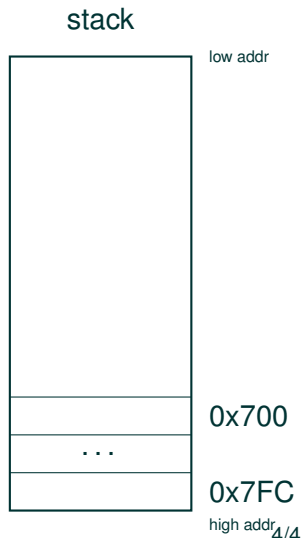
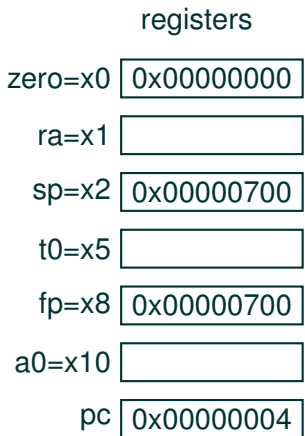
# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



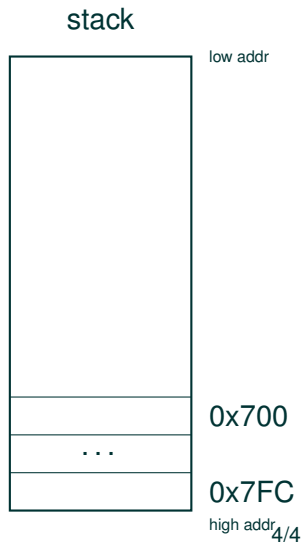
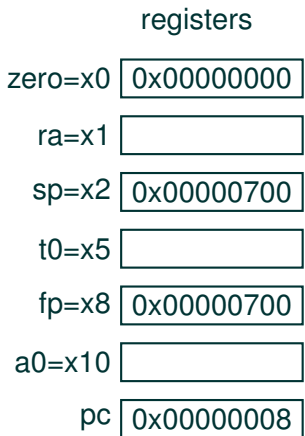
# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



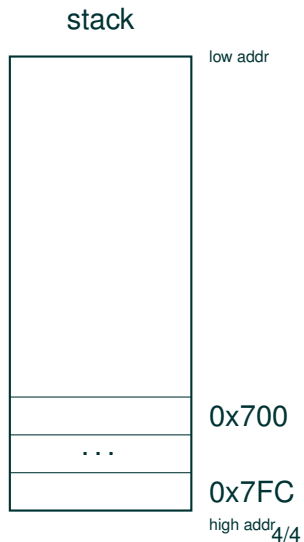
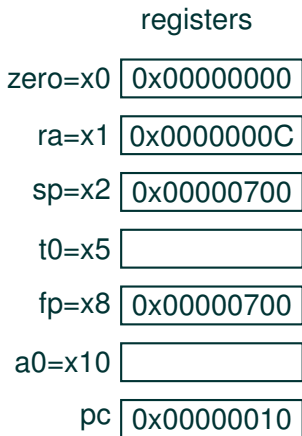
# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



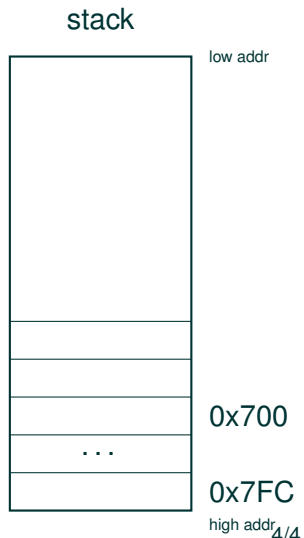
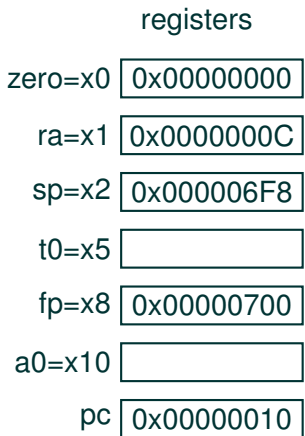
# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



# Buffer overflow

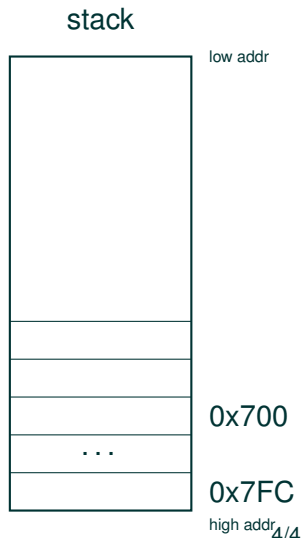
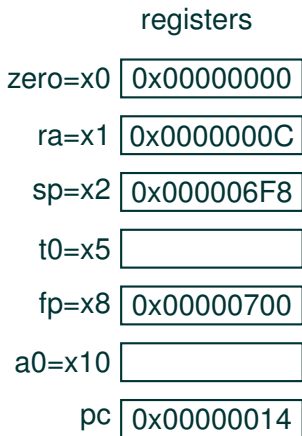
```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```





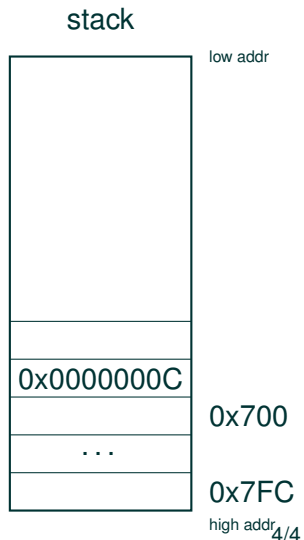
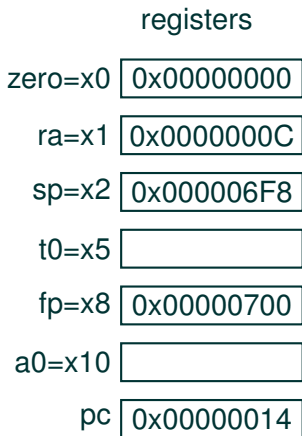
# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



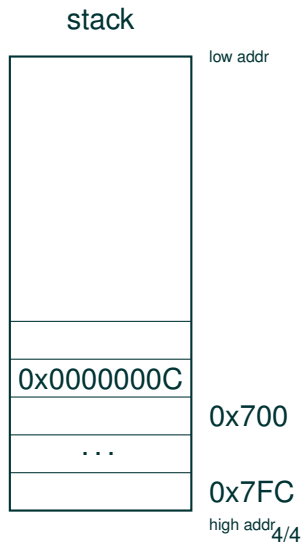
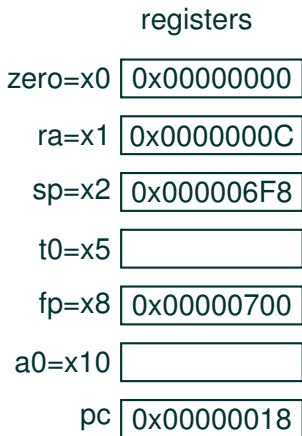
# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



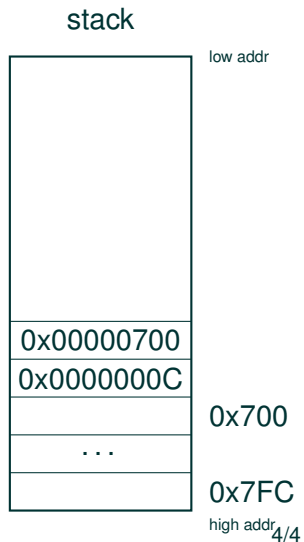
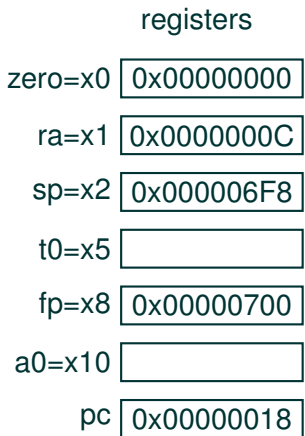
# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



# Buffer overflow

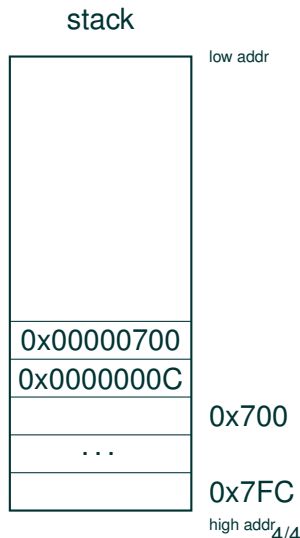
```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```



# Buffer overflow

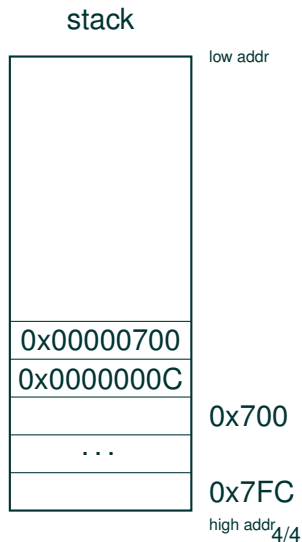
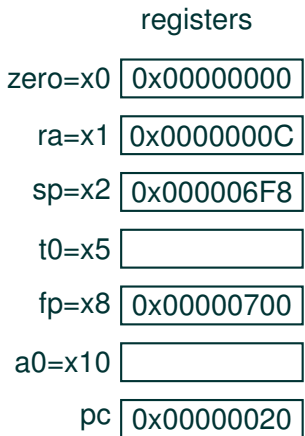
```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,0  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```

	registers
zero=x0	0x00000000
ra=x1	0x0000000C
sp=x2	0x000006F8
t0=x5	
fp=x8	0x00000700
a0=x10	
pc	0x0000001C



# Buffer overflow

```
_start:  
  ADDI sp,zero,0x700  
  ADDI fp,zero,0x700  
  JAL ra,main  
  EBREAK  
main:  
  ADDI sp,sp,-8  
  SW ra,4(sp)  
  SW fp,0(sp)  
  ADDI fp,sp,8  
  JAL ra,vuln  
  LW fp,0(sp)  
  LW ra,4(sp)  
  ADDI sp,sp,8  
  JALR zero,0(ra)
```





# Buffer overflow

vuln:

```
ADDI sp,sp,-24
SW ra,20(sp)
SW fp,16(sp)
ADDI fp,sp,24
ADDI a0,fp,-24
JAL ra,gets
LW ra,20(sp)
LW fp,16(sp)
ADDI sp,sp,24
JALR zero,0(ra)
```

gets:

```
LW t0,0x7FC(zero)
SW t0,0(a0)
ADDI a0,a0,4
BNE t0,zero,gets
JALR zero,0(ra)
```

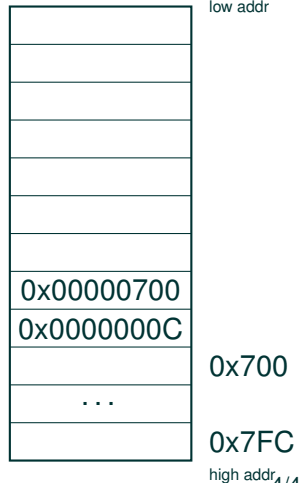
secret:

```
SW zero,0x7FC(zero)
EBREAK
```

registers

zero=x0	0x00000000
ra=x1	0x00000024
sp=x2	0x000006E0
t0=x5	
fp=x8	0x00000700
a0=x10	
pc	0x00000038

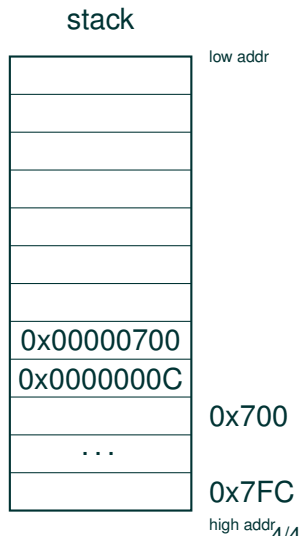
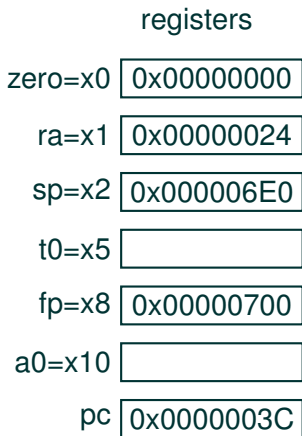
stack





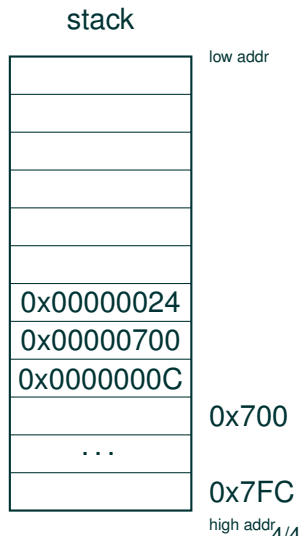
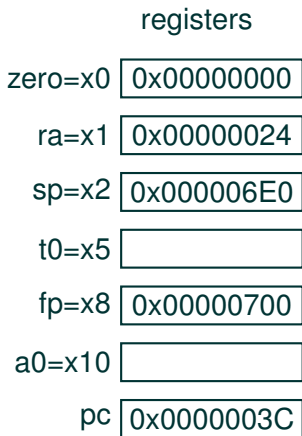
# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW ra,20(sp)
  SW fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL ra,gets
  LW ra,20(sp)
  LW fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW t0,0x7FC(zero)
  SW t0,0(a0)
  ADDI a0,a0,4
  BNE t0,zero,gets
  JALR zero,0(ra)
secret:
  SW zero,0x7FC(zero)
  EBREAK
```



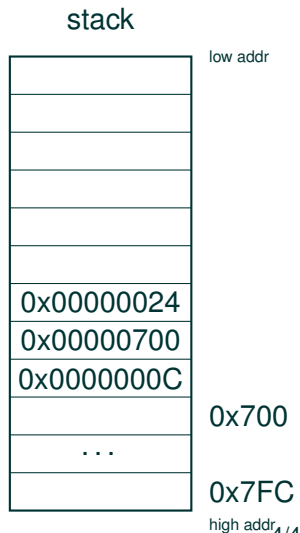
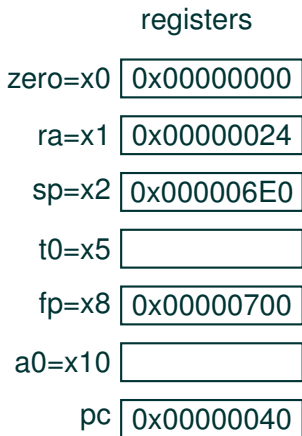
# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW ra,20(sp)
  SW fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL ra,gets
  LW ra,20(sp)
  LW fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW t0,0x7FC(zero)
  SW t0,0(a0)
  ADDI a0,a0,4
  BNE t0,zero,gets
  JALR zero,0(ra)
secret:
  SW zero,0x7FC(zero)
  EBREAK
```



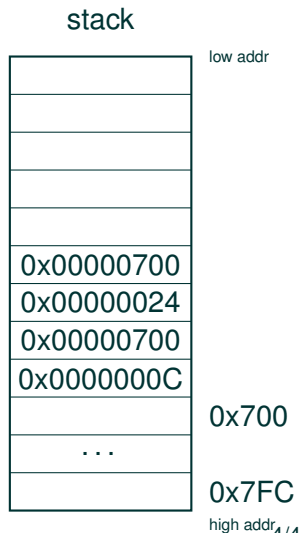
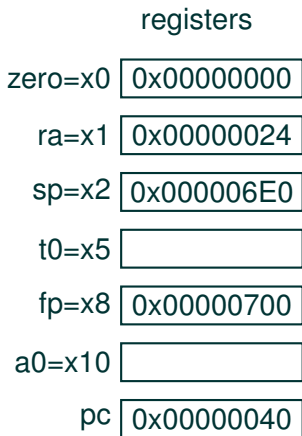
# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```



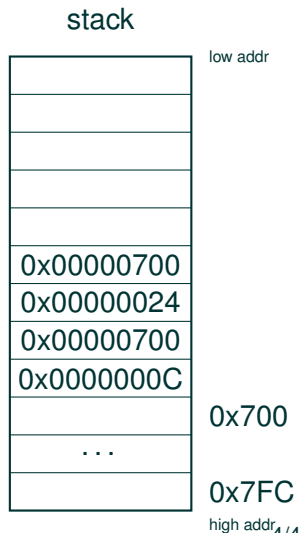
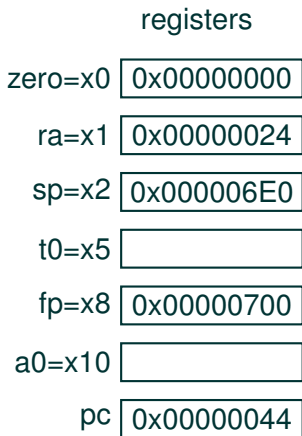
# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```



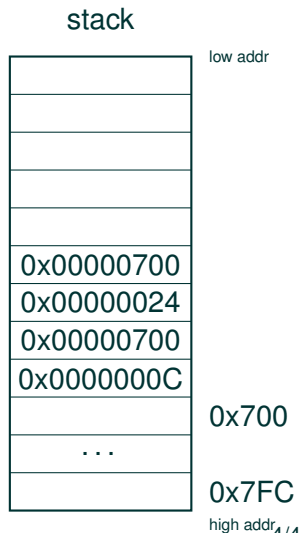
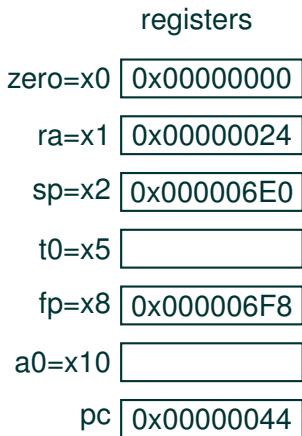
# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```



# Buffer overflow

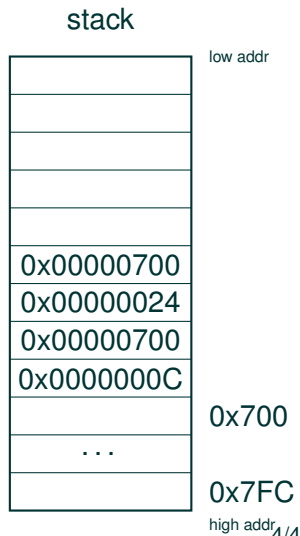
```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```



# Buffer overflow

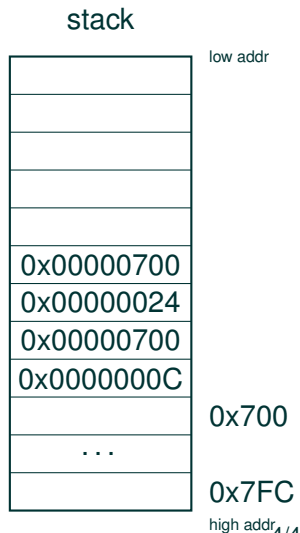
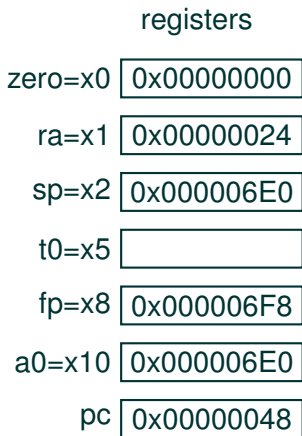
```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000024
sp=x2	0x000006E0
t0=x5	
fp=x8	0x000006F8
a0=x10	
pc	0x00000048



# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

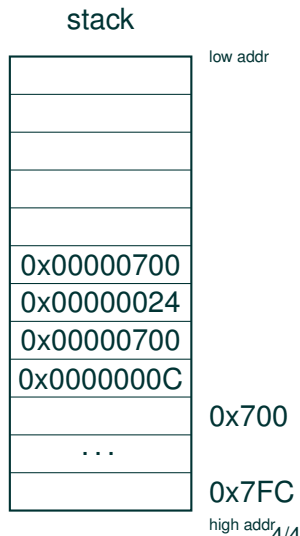




# Buffer overflow

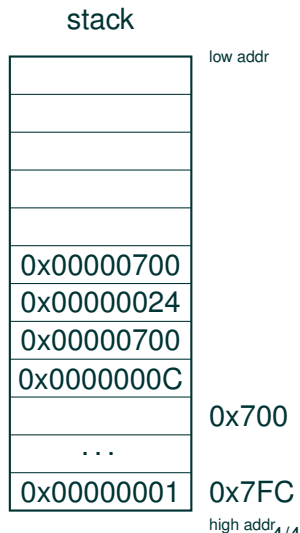
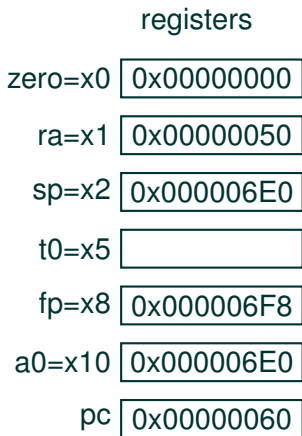
```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000024
sp=x2	0x000006E0
t0=x5	
fp=x8	0x000006F8
a0=x10	0x000006E0
pc	0x0000004C



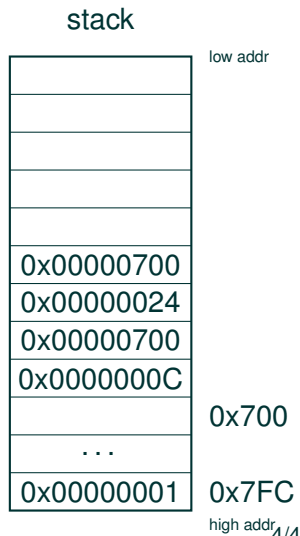
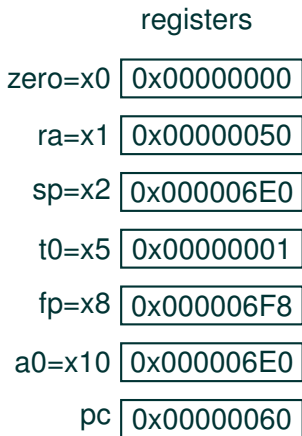
# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```



# Buffer overflow

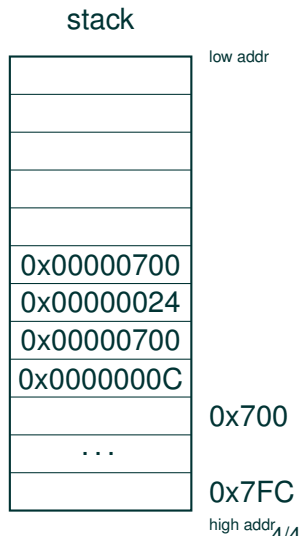
```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```



# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

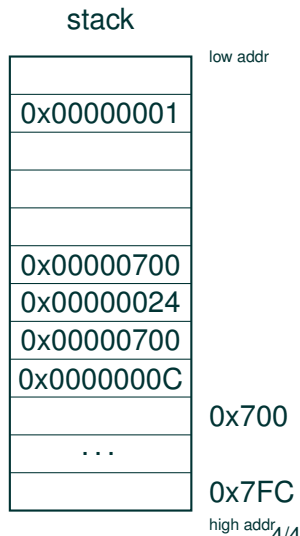
	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000001
fp=x8	0x000006F8
a0=x10	0x000006E0
pc	0x00000064



# Buffer overflow

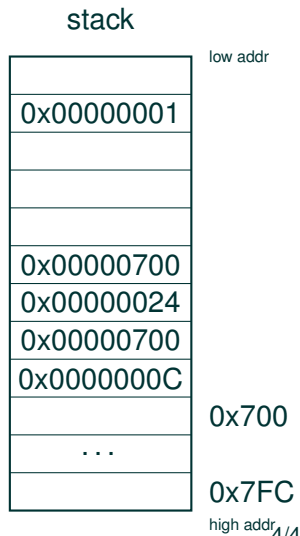
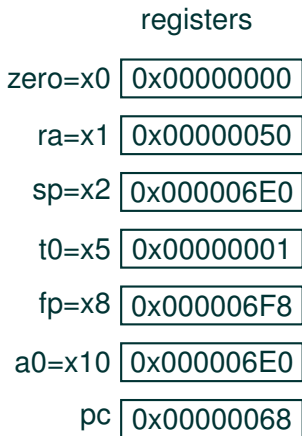
```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000001
fp=x8	0x000006F8
a0=x10	0x000006E0
pc	0x00000064



# Buffer overflow

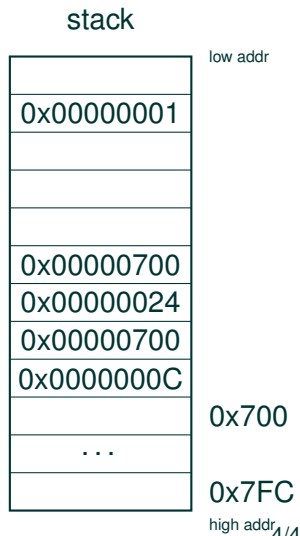
```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```



# Buffer overflow

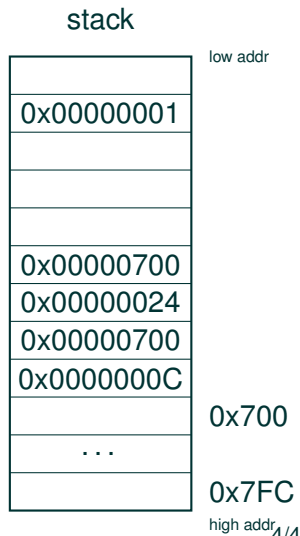
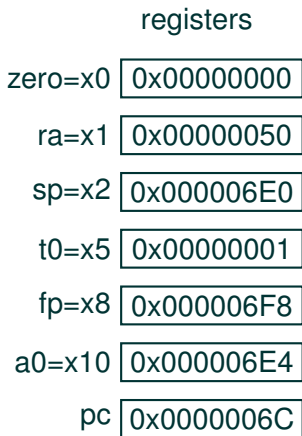
```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000001
fp=x8	0x000006F8
a0=x10	0x000006E4
pc	0x00000068



# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```





# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000001
fp=x8	0x000006F8
a0=x10	0x000006E4
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000002	0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

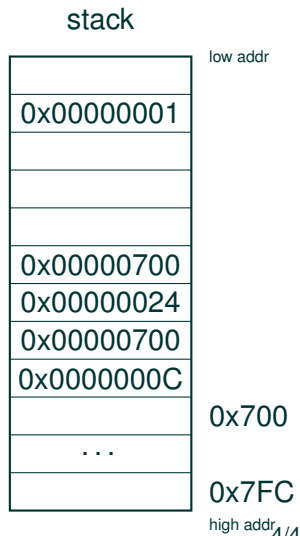
	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000002
fp=x8	0x000006F8
a0=x10	0x000006E4
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000002	0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000002
fp=x8	0x000006F8
a0=x10	0x000006E4
pc	0x00000064



# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

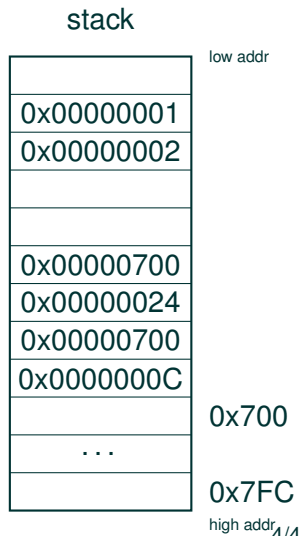
	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000002
fp=x8	0x000006F8
a0=x10	0x000006E4
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000002
fp=x8	0x000006F8
a0=x10	0x000006E4
pc	0x00000068



# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000002
fp=x8	0x000006F8
a0=x10	0x000006E8
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000002
fp=x8	0x000006F8
a0=x10	0x000006E8
pc	0x0000006C

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000002
fp=x8	0x000006F8
a0=x10	0x000006E8
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000003	0x7FC
		high addr



# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

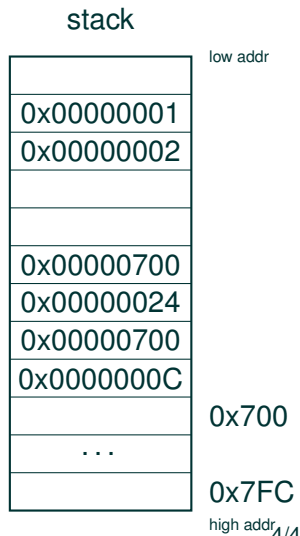
	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000003
fp=x8	0x000006F8
a0=x10	0x000006E8
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000003	0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000003
fp=x8	0x000006F8
a0=x10	0x000006E8
pc	0x00000064



# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000003
fp=x8	0x000006F8
a0=x10	0x000006E8
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000003
fp=x8	0x000006F8
a0=x10	0x000006E8
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000003
fp=x8	0x000006F8
a0=x10	0x000006EC
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000003
fp=x8	0x000006F8
a0=x10	0x000006EC
pc	0x0000006C

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000003
fp=x8	0x000006F8
a0=x10	0x000006EC
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000004	0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000004
fp=x8	0x000006F8
a0=x10	0x000006EC
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000004	0x7FC
		high addr



# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000004
fp=x8	0x000006F8
a0=x10	0x000006EC
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000004
fp=x8	0x000006F8
a0=x10	0x000006EC
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000004
fp=x8	0x000006F8
a0=x10	0x000006EC
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000004
fp=x8	0x000006F8
a0=x10	0x000006F0
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000004
fp=x8	0x000006F8
a0=x10	0x000006F0
pc	0x0000006C

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000004
fp=x8	0x000006F8
a0=x10	0x000006F0
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000005	0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000005
fp=x8	0x000006F8
a0=x10	0x000006F0
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000005	0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000005
fp=x8	0x000006F8
a0=x10	0x000006F0
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000700	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr



# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000005
fp=x8	0x000006F8
a0=x10	0x000006F0
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000005
fp=x8	0x000006F8
a0=x10	0x000006F0
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000005
fp=x8	0x000006F8
a0=x10	0x000006F4
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000005
fp=x8	0x000006F8
a0=x10	0x000006F4
pc	0x0000006C

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000005
fp=x8	0x000006F8
a0=x10	0x000006F4
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000070	0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000070
fp=x8	0x000006F8
a0=x10	0x000006F4
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000070	0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000070
fp=x8	0x000006F8
a0=x10	0x000006F4
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000024	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000070
fp=x8	0x000006F8
a0=x10	0x000006F4
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr



# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000070
fp=x8	0x000006F8
a0=x10	0x000006F4
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000070
fp=x8	0x000006F8
a0=x10	0x000006F8
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000070
fp=x8	0x000006F8
a0=x10	0x000006F8
pc	0x0000006C

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000070
fp=x8	0x000006F8
a0=x10	0x000006F8
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000000	0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006F8
pc	0x00000060

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000700	
	0x0000000C	
		0x700
	...	
	0x00000000	0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006F8
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000700	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006F8
pc	0x00000064

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006F8
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr



# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006FC
pc	0x00000068

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006FC
pc	0x0000006C

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006FC
pc	0x00000070

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000050
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006FC
pc	0x00000050

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000070
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006FC
pc	0x00000050

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000070
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x000006F8
a0=x10	0x000006FC
pc	0x00000054

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000070
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x00000005
a0=x10	0x000006FC
pc	0x00000054

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr

# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```

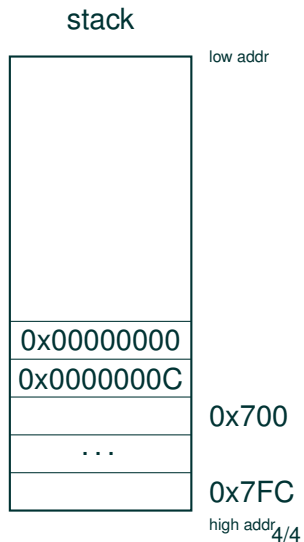
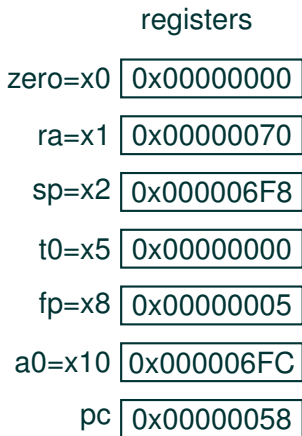
	registers
zero=x0	0x00000000
ra=x1	0x00000070
sp=x2	0x000006E0
t0=x5	0x00000000
fp=x8	0x00000005
a0=x10	0x000006FC
pc	0x00000058

	stack	
		low addr
	0x00000001	
	0x00000002	
	0x00000003	
	0x00000004	
	0x00000005	
	0x00000070	
	0x00000000	
	0x0000000C	
		0x700
	...	
		0x7FC
		high addr



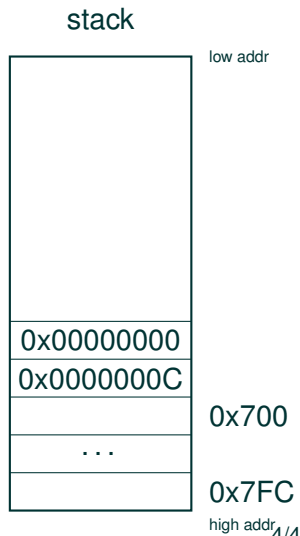
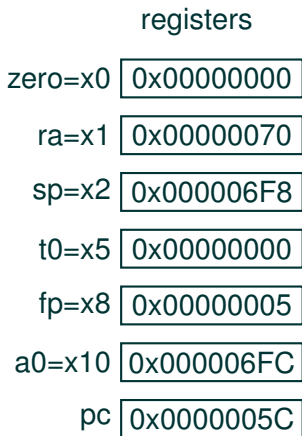
# Buffer overflow

```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```



# Buffer overflow

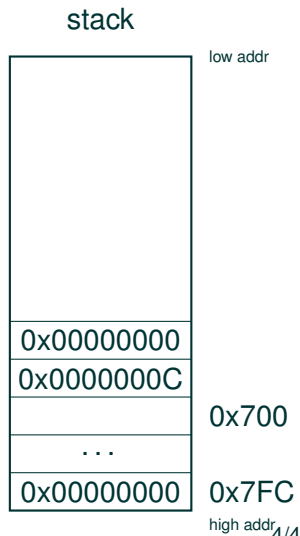
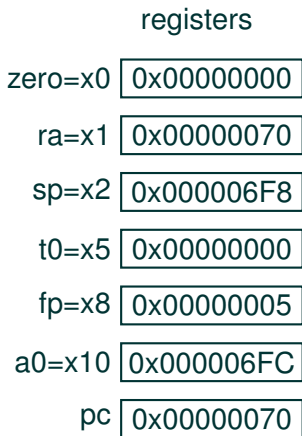
```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```





# Buffer overflow

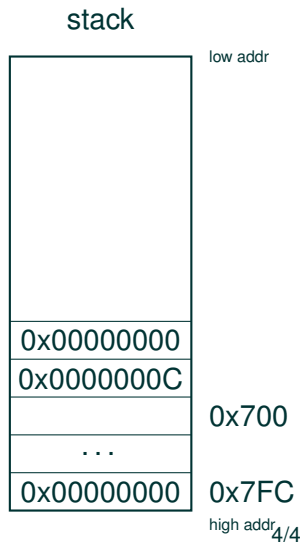
```
vuln:  
  ADDI sp,sp,-24  
  SW   ra,20(sp)  
  SW   fp,16(sp)  
  ADDI fp,sp,24  
  ADDI a0,fp,-24  
  JAL  ra,gets  
  LW   ra,20(sp)  
  LW   fp,16(sp)  
  ADDI sp,sp,24  
  JALR zero,0(ra)  
gets:  
  LW   t0,0x7FC(zero)  
  SW   t0,0(a0)  
  ADDI a0,a0,4  
  BNE  t0,zero,gets  
  JALR zero,0(ra)  
secret:  
  SW   zero,0x7FC(zero)  
  EBREAK
```



# Buffer overflow

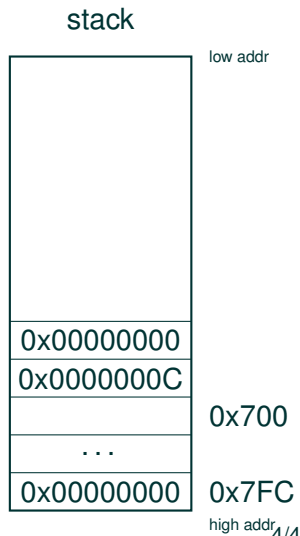
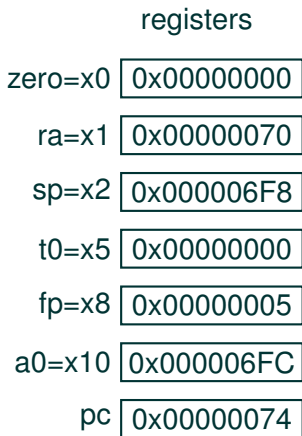
```
vuln:
  ADDI sp,sp,-24
  SW ra,20(sp)
  SW fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL ra,gets
  LW ra,20(sp)
  LW fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW t0,0x7FC(zero)
  SW t0,0(a0)
  ADDI a0,a0,4
  BNE t0,zero,gets
  JALR zero,0(ra)
secret:
  SW zero,0x7FC(zero)
  EBREAK
```

	registers
zero=x0	0x00000000
ra=x1	0x00000070
sp=x2	0x000006F8
t0=x5	0x00000000
fp=x8	0x00000005
a0=x10	0x000006FC
pc	0x00000074



# Buffer overflow

```
vuln:
  ADDI sp,sp,-24
  SW   ra,20(sp)
  SW   fp,16(sp)
  ADDI fp,sp,24
  ADDI a0,fp,-24
  JAL  ra,gets
  LW   ra,20(sp)
  LW   fp,16(sp)
  ADDI sp,sp,24
  JALR zero,0(ra)
gets:
  LW   t0,0x7FC(zero)
  SW   t0,0(a0)
  ADDI a0,a0,4
  BNE  t0,zero,gets
  JALR zero,0(ra)
secret:
  SW   zero,0x7FC(zero)
  EBREAK
```



# Function Calls & Stack examples

---

Stefan Mangard

November 10, 2021

Computer Organization and Networks  
Graz University of Technology