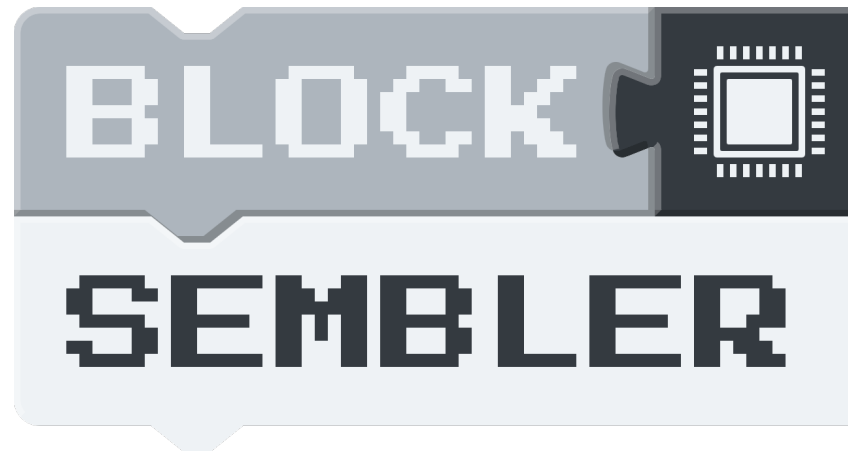
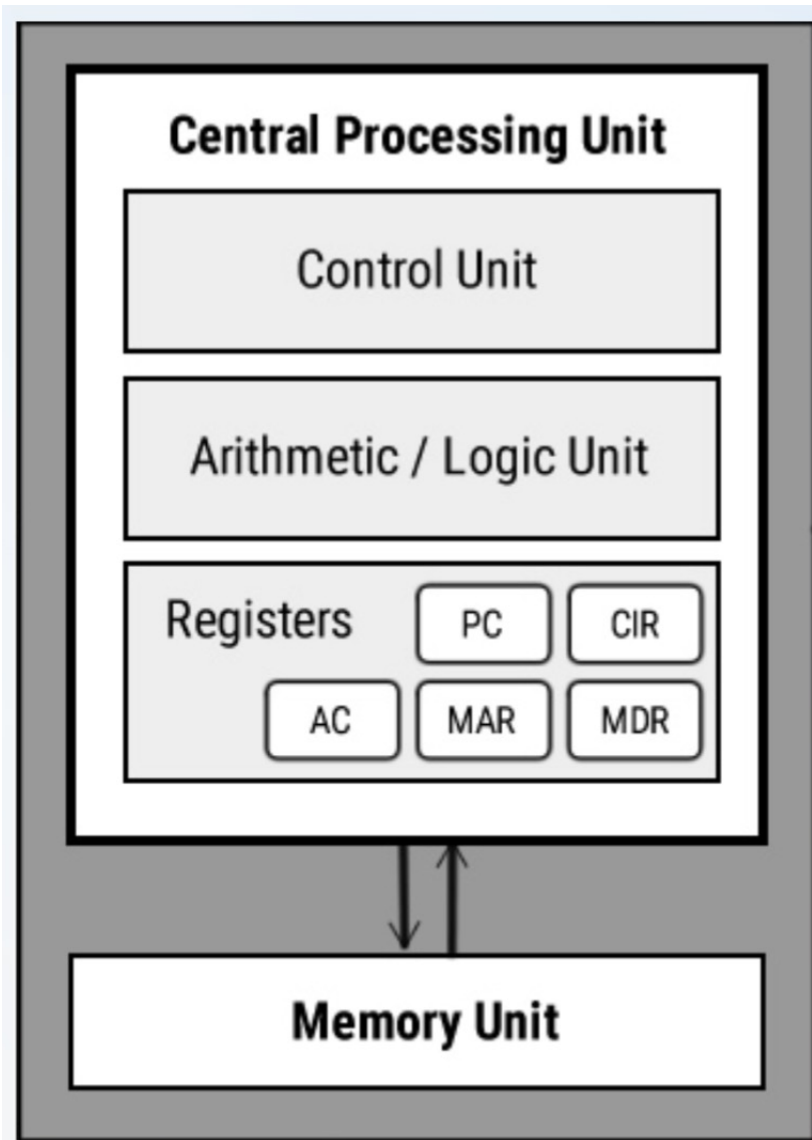


Assembly Programming with



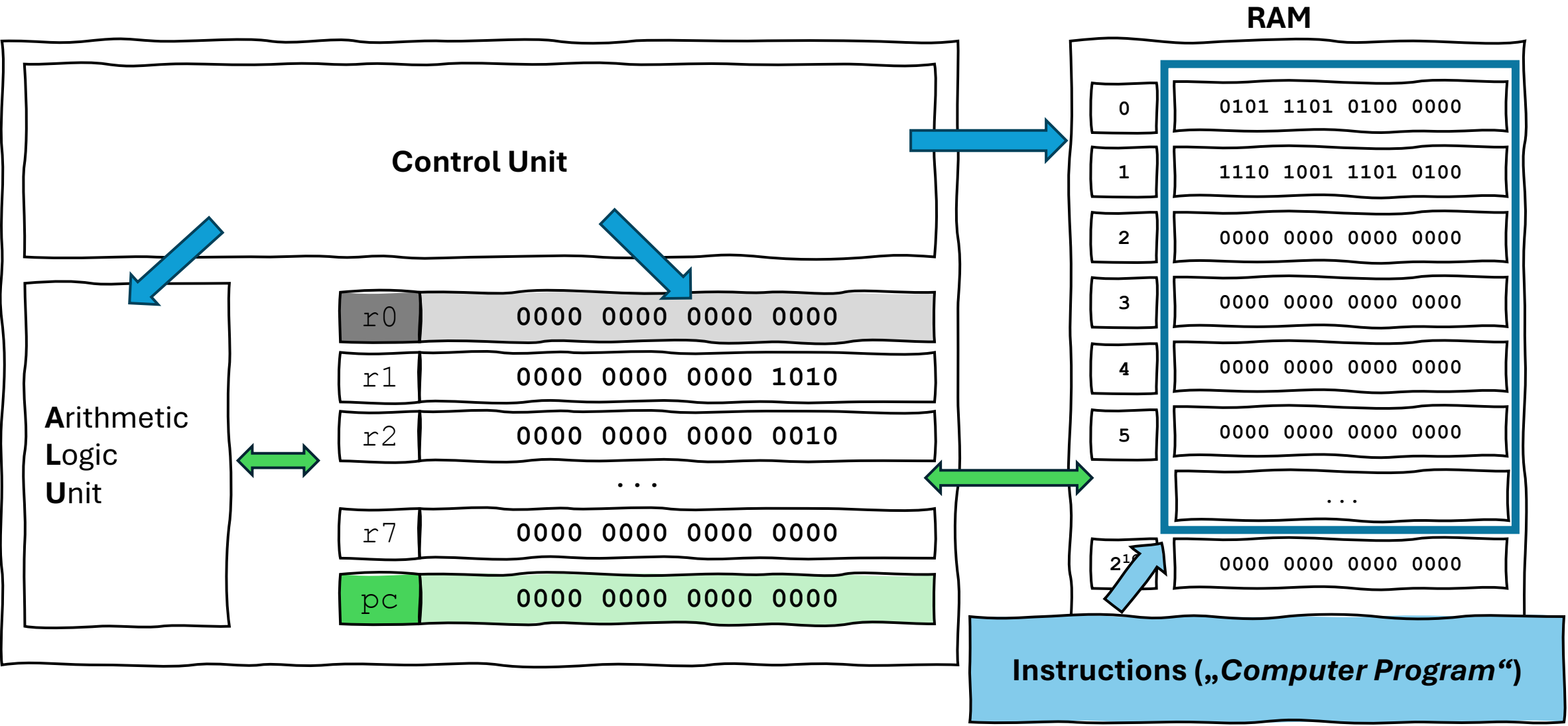
Recap



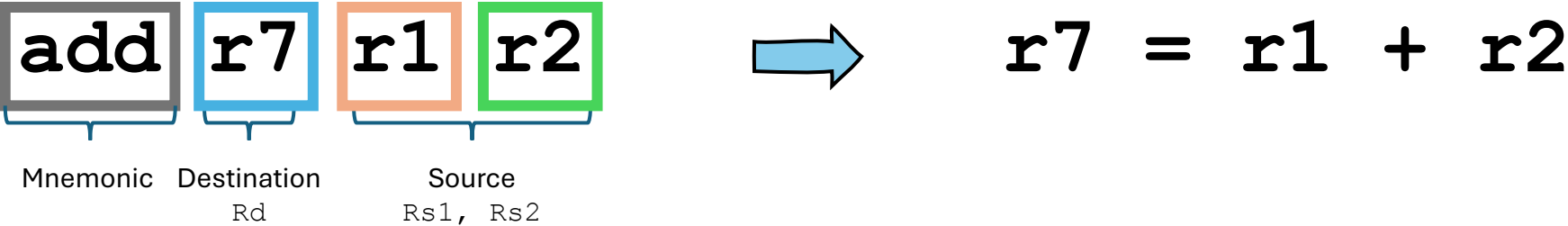
ANNA (A New Noncomplex Architecture)

- 16-bit Architecture
- 8 Registers (r_0, \dots, r_7)
- 16 Instructions
- 2^{16} x Words of RAM

ANNA Architecture (Overview)

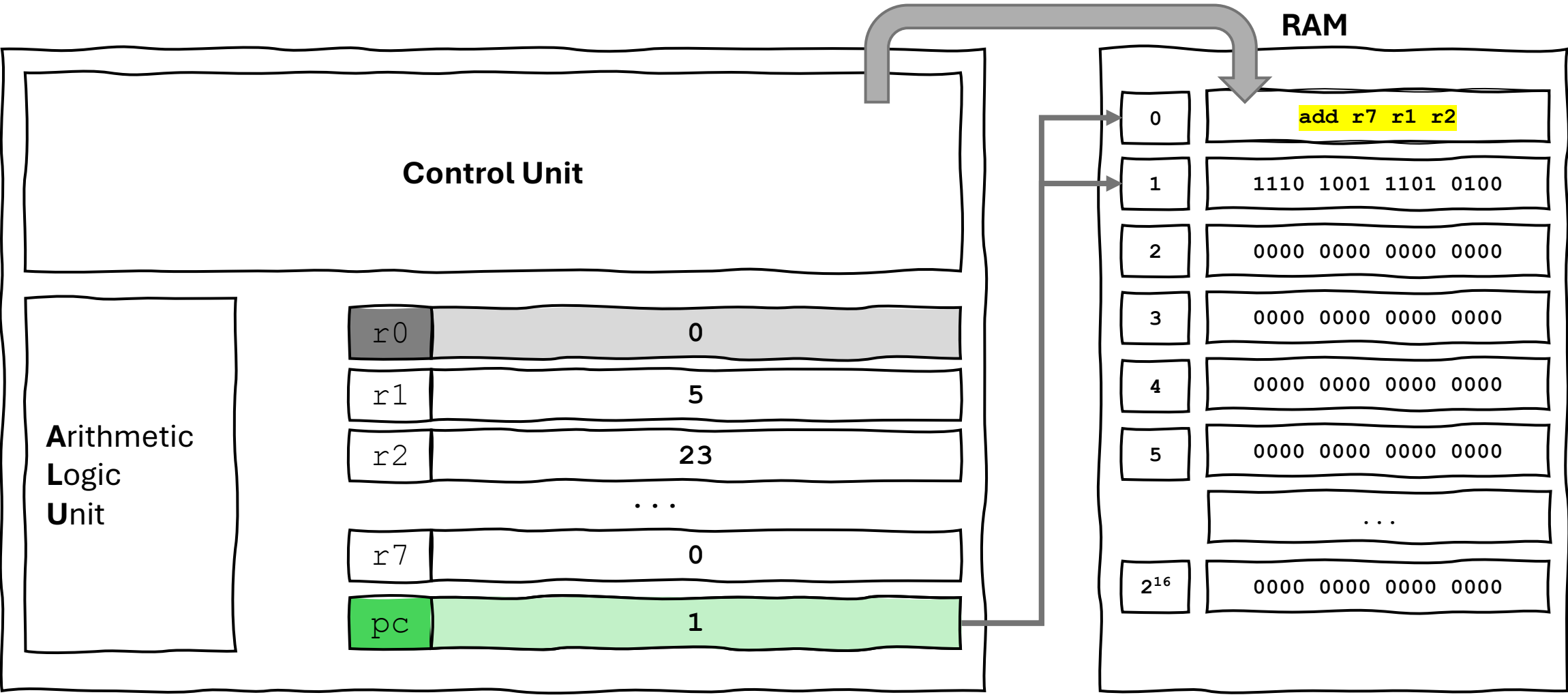


ADD Instruction

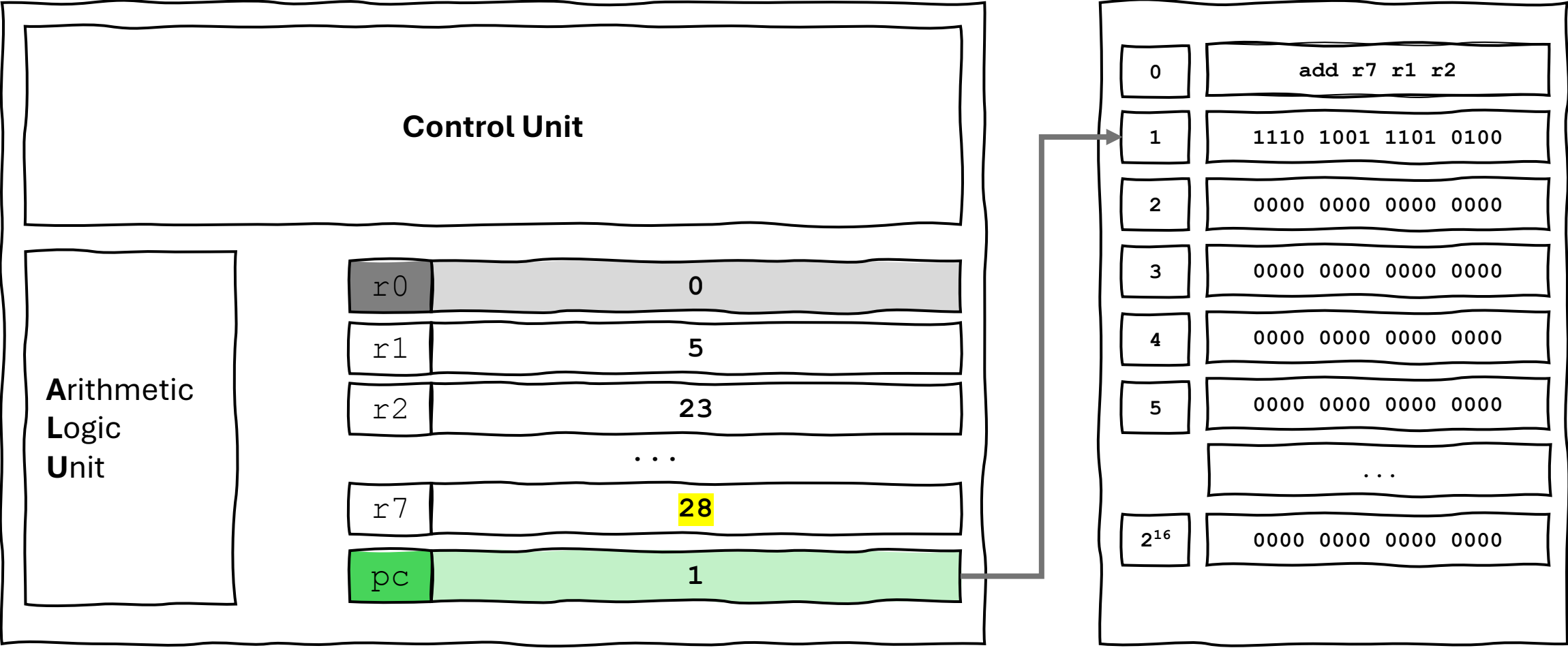


0	0	0	0	1	1	1	0	0	1	0	1	0	0	0	0
15			12	11		9	8		6	5		3	2		0
Opcode				Rd			Rs ₁			Rs ₂			Unused		

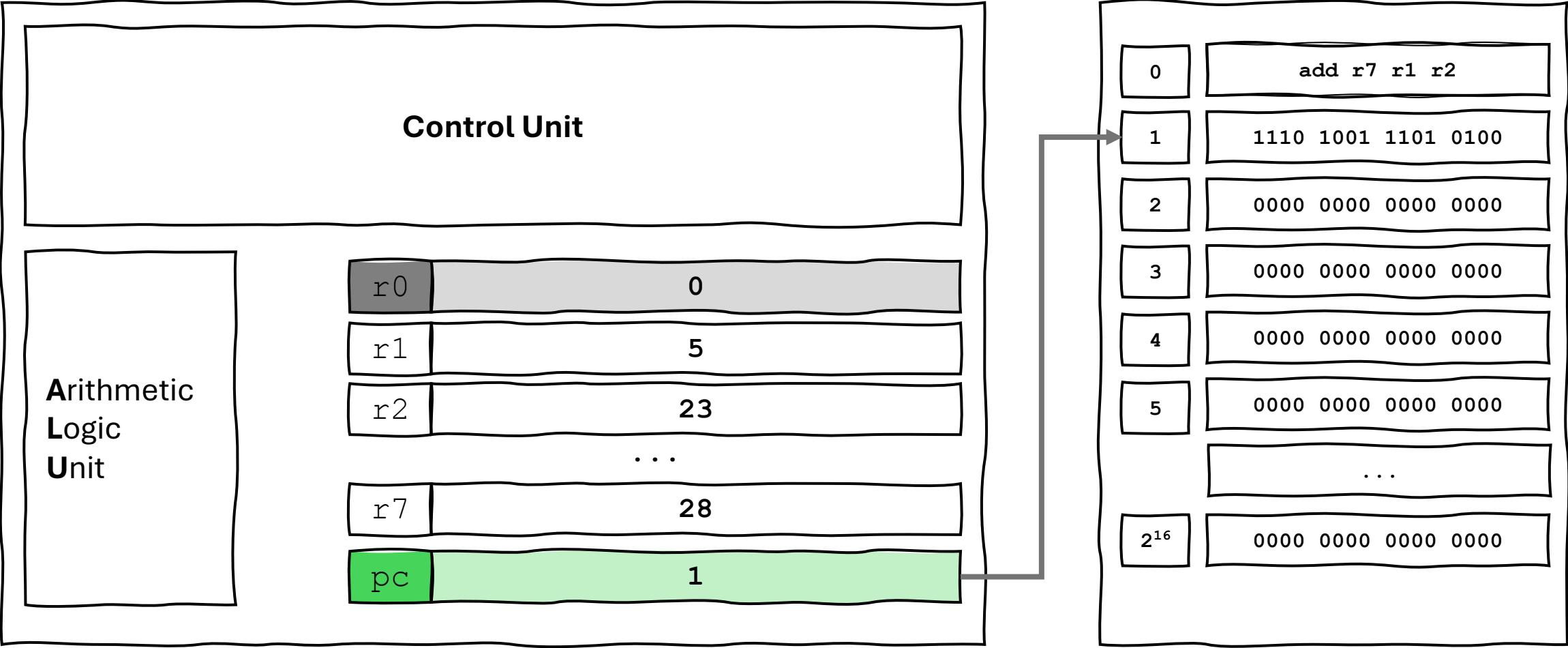
ANNA Architecture (Overview)



ANNA Architecture (Overview)




ANNA Architecture (Overview)

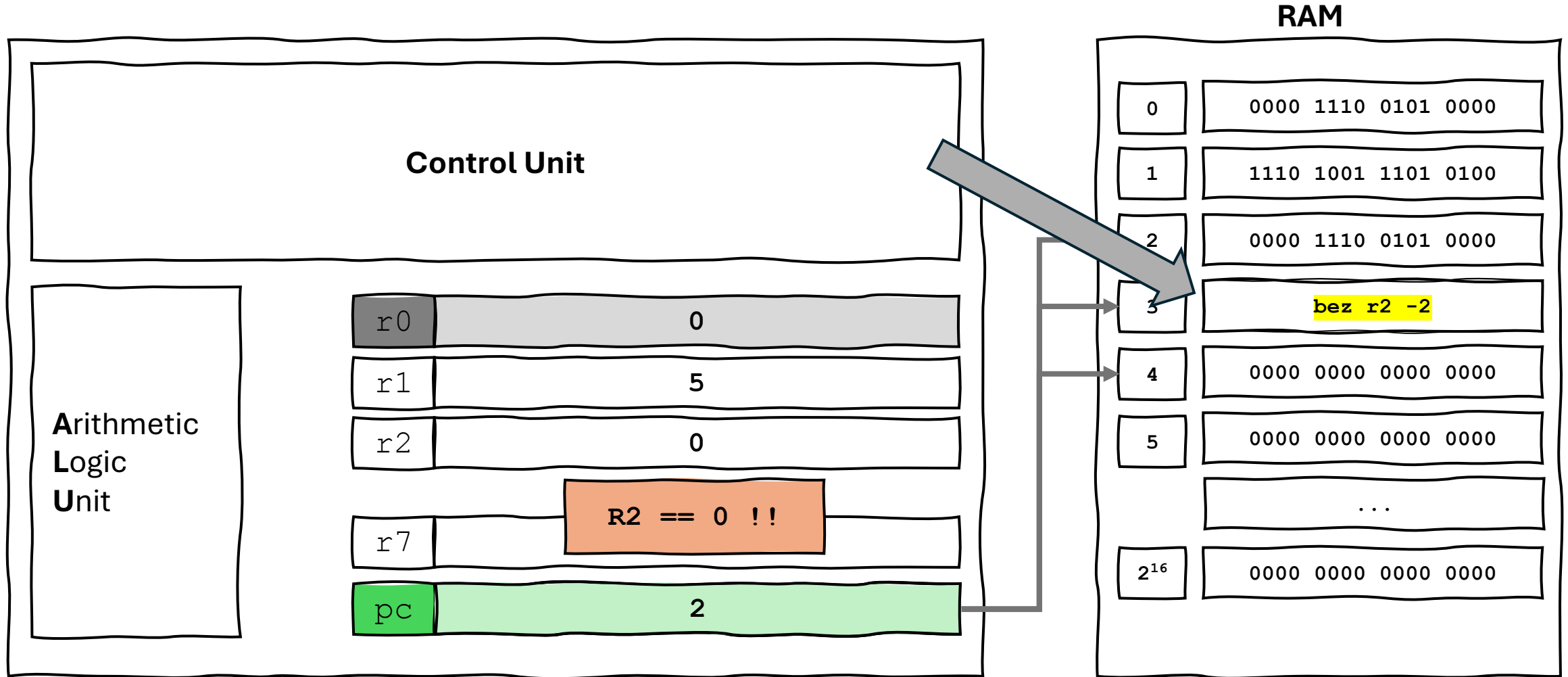


Assembly Code Example

```
add r7 r2 r3  
shf r3 r1 4  
and r2 r1 r3  
or  r5 r6 r7
```




ANNA Architecture (Overview)



Jump Example

```
loop:  addi r1 r1 -1  
        out r1  
        bgz r1 -2  
  
        .halt
```



Anna Instruction Set

(Overview)

- add
- sub
- and
- or
- not
- shf *(shift)*
- lli *(load lower immediate)*
- lui *(load upper immediate)*
- lw *(load word)*
- sw *(store word)*
- bez *(branch equal zero)*
- bgz *(branch greater zero)*
- addi *(add immediate)*
- jalr *(jump and link register)*
- in
- out

Blocksembler

The screenshot displays the Blocksembler web application interface. The browser address bar shows the URL `blocksembler.eden.univie.ac.at`. The application has a dark-themed header with a menu bar containing `File`, `Edit Code`, `Run Code`, and `Settings`. On the left, a sidebar lists navigation options: `Program Structure`, `Primitives`, `System Instructions`, `Memory Instructions`, `Arithmetic Instructions`, `Logic Instructions`, and `Compare and Branching`.

The main workspace is divided into two panels. The left panel shows a block-based assembly editor with the following structure:

- ENTRYPOINT** (blue label)
- `move value` block: `hex value: 0x0018` to register `Register: $ 1`
- `Register: $ 2 := Register: $ 1` block with a dropdown set to `-` and `decimal value: 1`
- `label: @ loop` (blue label)
- `compare` block: `Register: $ 2` and `decimal value: 1` and update status register
- `jump to address` block: `Label: > isPrime` if last comparison was `equal`
- `move value` block: `Register: $ 1` to register `Register: $ 3`
- `label: @ innerLoop` (blue label)
- `compare` block: `Register: $ 3` and `decimal value: 0` and update status register
- `jump to address` block: `Label: > exitInner` if last comparison was `less or equal`
- `Register: $ 3 := Register: $ 3` block with a dropdown set to `-` and `Register: $ 2`
- `jump to address` block: `Label: > innerLoop`
- `label: @ exitInner` (blue label)
- `jump to address` block: `Label: > noPrime` if last comparison was `equal`
- `Register: $ 2 := Register: $ 2` block with a dropdown set to `-` and `decimal value: 1`

The right panel shows the corresponding assembly code in a text-based view:

```
1    mov $1, 0x0018
2    sub $2, $1, 1
3
4 @loop:
5    cmp $2, 1
6    beq >isPrime
7    mov $3, $1
8
9 @innerLoop:
10   cmp $3, 0
11   ble >exitInner
12   sub $3, $3, $2
13   jmp >innerLoop
14
15 @exitInner:
16   beq >noPrime
17   sub $2, $2, 1
18   jmp >loop
19
20 @noPrime:
21   mov $4, 1
22   jmp >exit
23
24 @isPrime:
25   mov $4, 0
```

Blocksembler Instructions

add  and  and store result to 

Start/Halt

- Blocksembler programs always begin with a **start** block.
- When the Control Unit encounters a **halt** instruction (represented by the *Halt* block), the program terminates.
- Using multiple *Halt* blocks within a program is allowed.



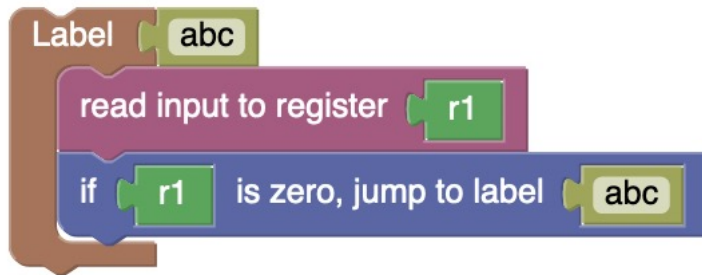
JUMP/Branch Instructions

There are instructions that cause the program flow to jump to a specific line in the program when a certain condition is met.



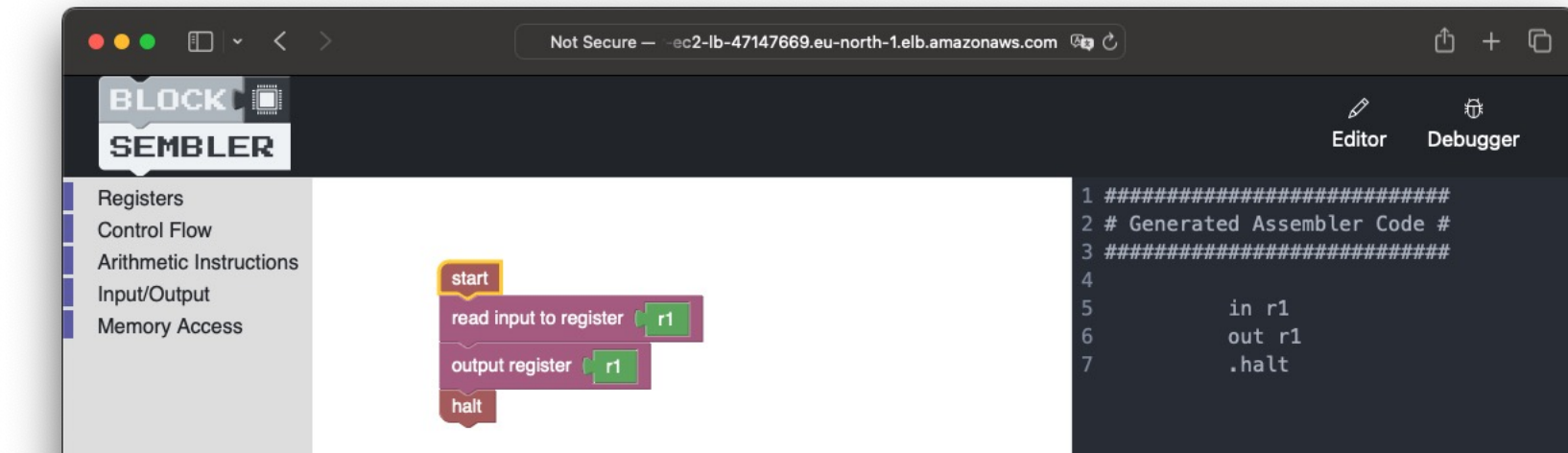
Label

To define the target of such a jump, `labels` are used.



```
1 #####
2 # Generated Assembler Code #
3 #####
4
5
6 abc:      in r1
7           bez r1 &abc
```

Live Demonstration



Resources

Presentation:

<https://blocksembler.github.io/presentation.pdf>

ANNA Documentation:

<https://blocksembler.github.io/anna.pdf>

Blocksembler:

<https://blocksembler.eden.univie.ac.at>

mail: `florian.woerister@univie.ac.at`

