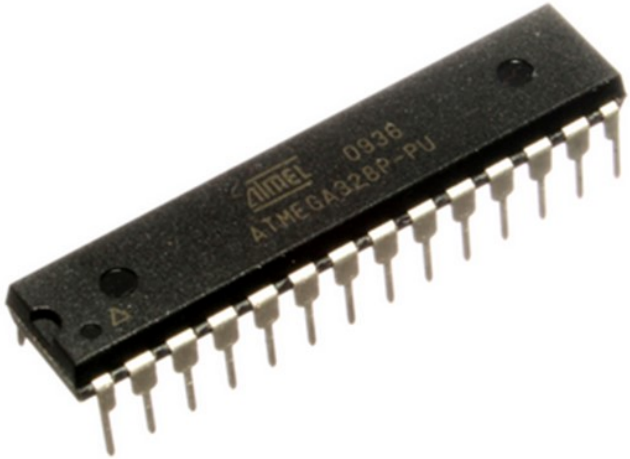
# AVR ATMEGA328P-PU




© Prof. Engº esp Luiz Antonio Vargas Pinto
www.vargasp.com
© setembro 2026

# Histórico

- A arquitetura básica foi desenvolvida na Noruega pelos doutorandos Alf-Egil-Boden e Vegard Wollan da *University of Science and Technology* em **1992**
- Sugere-se, porém nunca foi comprovado, que AVR é o acrônimo de **A**lf, **V**egard e **R**ISC
- Após algum tempo de aperfeiçoamento, eles venderam o projeto para a ATMEL
- O primeiro AVR, o AT90S1200, foi lançado em **1997**.
- A ATMEL e a tecnologia AVR foi comprada pela Microchip em **2016**

© Prof. Vargas

# Algumas considerações

- Aplicações
  - Câmeras, Aparelhos de DVD, Telefones celulares, Microondas, Máquinas de lavar roupa, etc . . .

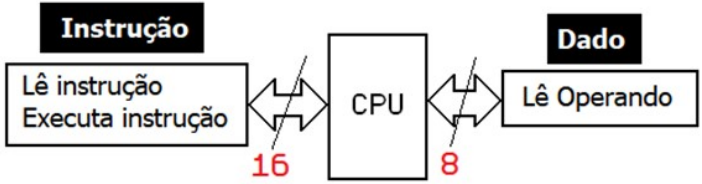
- MicroProcessador x MicroControlador
  - Microprocessadores tem periféricos externos ao chip
  - Microcontroladores tem os periféricos internos ao chip
    - muitas vezes ele é citado como:

      Computador em um chip

© Prof. Vargas

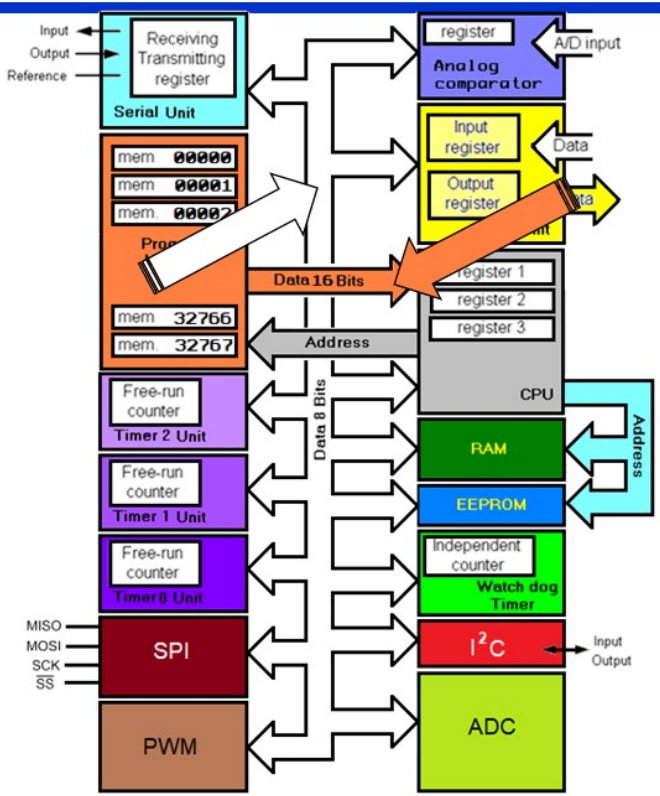
# Arquitetura Harvard

- Data bus independe do instruction bus
  - Permite que a CPU acesse aos dois bus simultaneamente
    - Dados e instruções simultaneamente
  - Melhora o desempenho
- Memória e dados tem tamanhos diferentes
  - AVR Atmega328p: instruções 16 bits / dados 8 bits




© Prof. Vargas

# Detalhe




© Prof. Vargas

# RISC

- As instruções são maiores e portanto mais poderosas (16 bits)
- instruções são mais eficientes, mesmo a 1 μs (embora necessite de um cristal de 16 MHz)

© Prof. Vargas

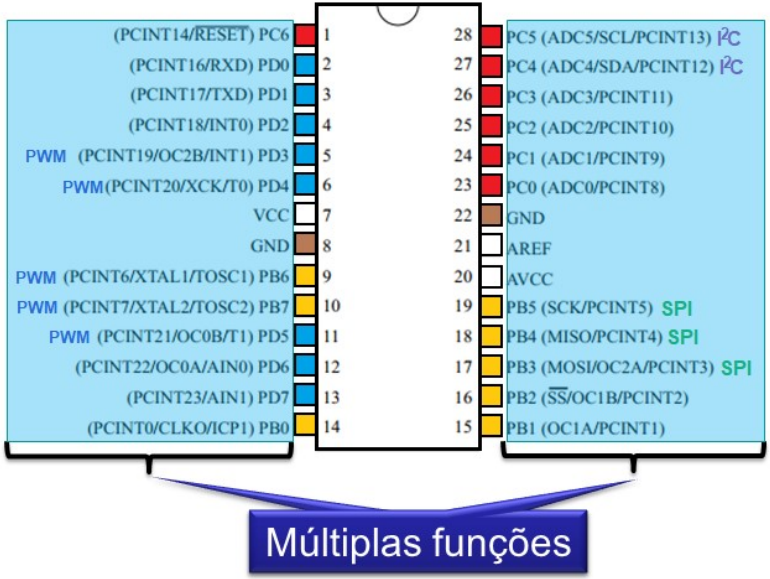
# O ATMEGA328p-pu

- 32768 endereços de programa em FlashROM
- 2048 bytes de memória SRAM
- 1024 bytes de memória EEPROM
- 1 Canal serial USART
- 2 Canais SPI
- 1 Canal $I^2C$
- 3 Timers programáveis por software
- 1 Timer Watchdog
- 23 ports de I/O (PB, PC e PD)
- 1 Comparador Analógico
- 6 Canais de PWM
- 8 Conversores A/D 10 bits
- 131 instruções

© Prof. Vargas

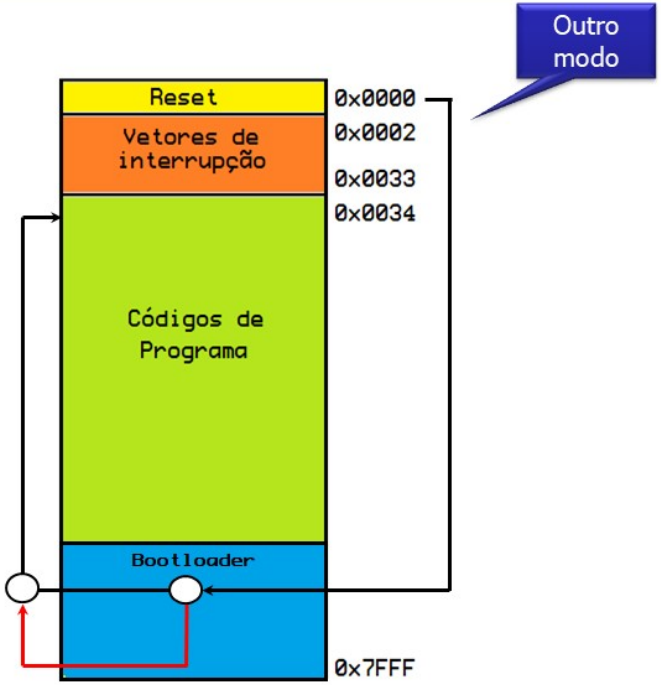
# Overview

- Funciona de 1.8 a 5.5V
- Flashrom pode ser gravada pelo menos 10.000 vezes
- EEPROM permite pelo menos 100.000 gravações
- Dados retidos por 20 anos a 85ºC ou 100 anos a 25ºC




© Prof. Vargas

# Estrutura da FlashRom



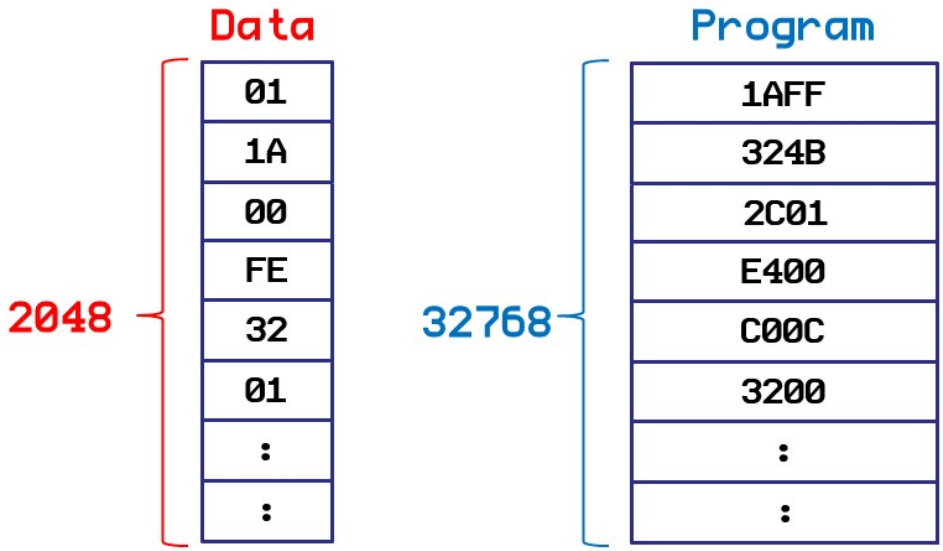

© Prof. Vargas

# Arquitetura de memória

© Prof. Vargas

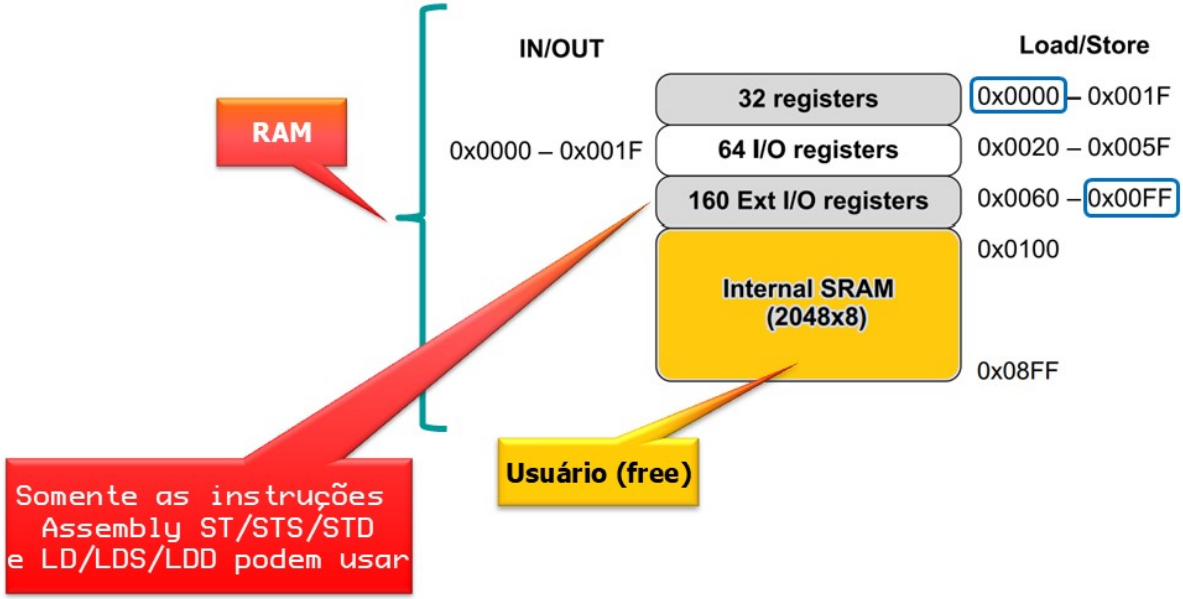
# Memória

No caso de microcontroladores, **a área de dados é composta por bytes**. A área de programas é maior podendo armazenar instruções com 12, 14, **16** e 32 bits




© Prof. Vargas

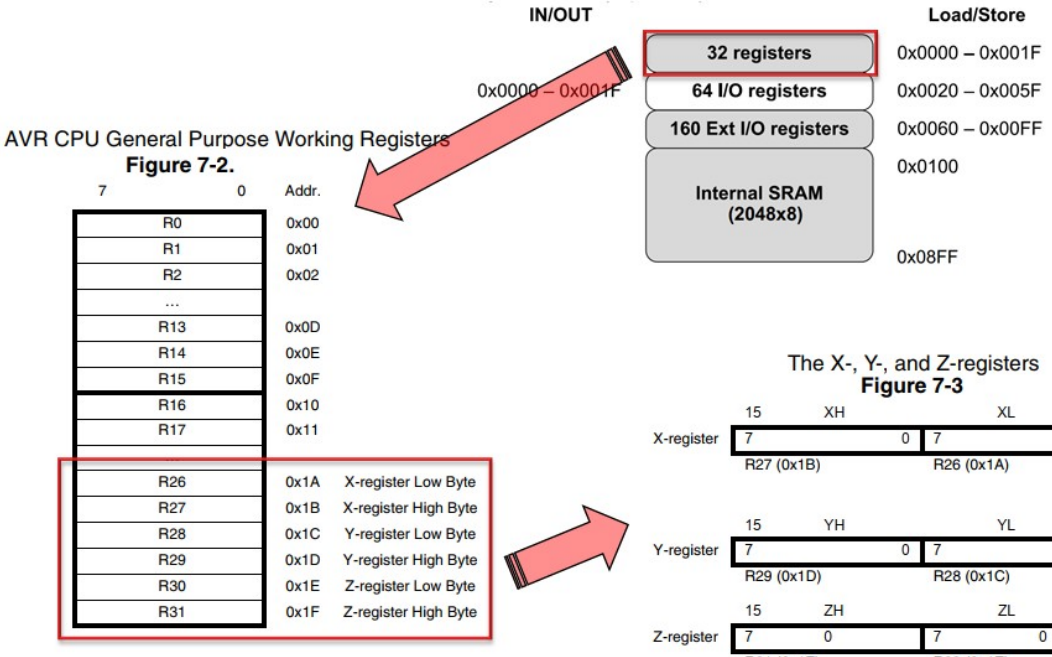
# Registros de controle e Memória de dados



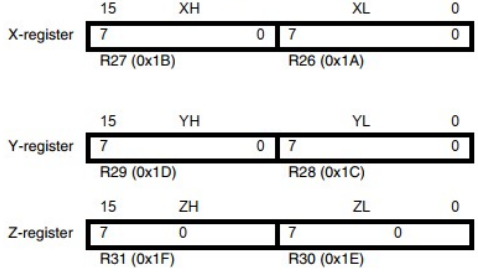

© Prof. Vargas

# Os registradores

AVR CPU General Purpose Working Registers
**Figure 7-2.**

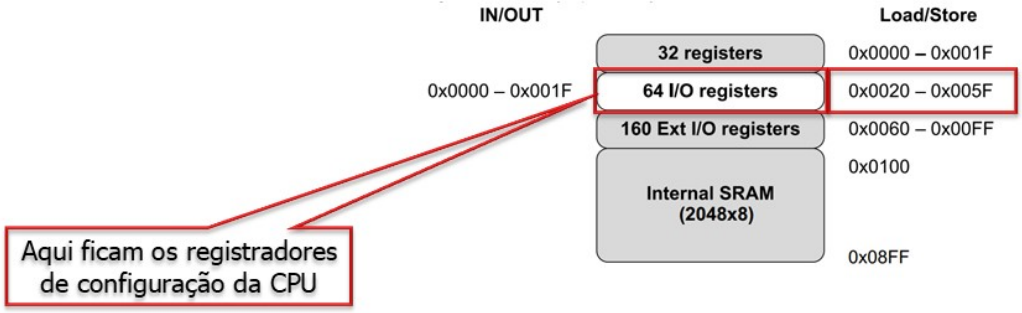



The X-, Y-, and Z-registers
**Figure 7-3**




© Prof. Vargas

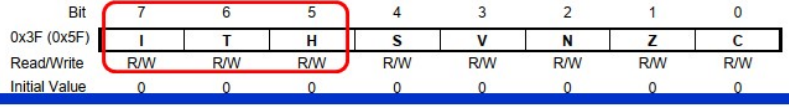
# 64 I/O Registers




© Prof. Vargas

# Status REGister – SREG

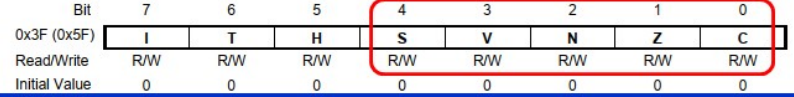
© Prof. Vargas



| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x3F (0x5F) | I | T | H | S | V | N | Z | C |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- Bit 7: I Global Interrupt Enable
  - 1 = Habilita  0 = Desabilita
- Bit 6: T Copy Storage
  - Usados em operações de bit.
- Bit 5: H Half Carry Flag
  - Usado em operações matemáticas
    - é o bit excedente

© Prof. Vargas



| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x3F (0x5F) | I | T | H | S | V | N | Z | C |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 4: S Sign**
  - O bit S é sempre um exclusive OR entre o flag N e o complemento de 2 do bit V
- **Bit 3: V Overflow do complemento de 2**
  - Ele suporta aritmética em complemento de 2
- **Bit 2: N Negative Flag**
  - Indica resultado negativo em uma operação lógica ou aritmética
- **Bit 1: Z Zero Flag**
  - Indica ocorrência de Zero em alguma operação
- **Bit 0: C Carry Flag**
  - Usado em operações matemáticas – é o bit excedente

© Prof. Vargas

# MCU Status REGister

© Prof. Vargas

MCU Status Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | WDRF | BORF | EXTRF | PORF |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

- Bit 3: WDRF Watchdog System Reset Flag
  - 1 = Ocorreu Watchdog por falha de CPU
  - 0 = Ocorreu Power-On Reset ou escrita de 0
- Bit 2: BORF Brown-Out Reset Flag
  - 1 = Ocorreu Brown-Out
  - 0 = Ocorreu Power-On Reset ou escrita de 0
- Bit 1: EXTRF EXternal Reset Flag
  - 1 = Ocorreu
  - 0 = Não ocorreu ou escrita de 0
- Bit 0: PORF Power-On Reset Flag
  - 1 = Ocorreu Power-On Reset
  - 0 = Somente por escrita de 0

© Prof. Vargas

# ESTUDO DAS INSTRUÇÕES

AVR ATMEGA328p-pu

© Prof. Vargas

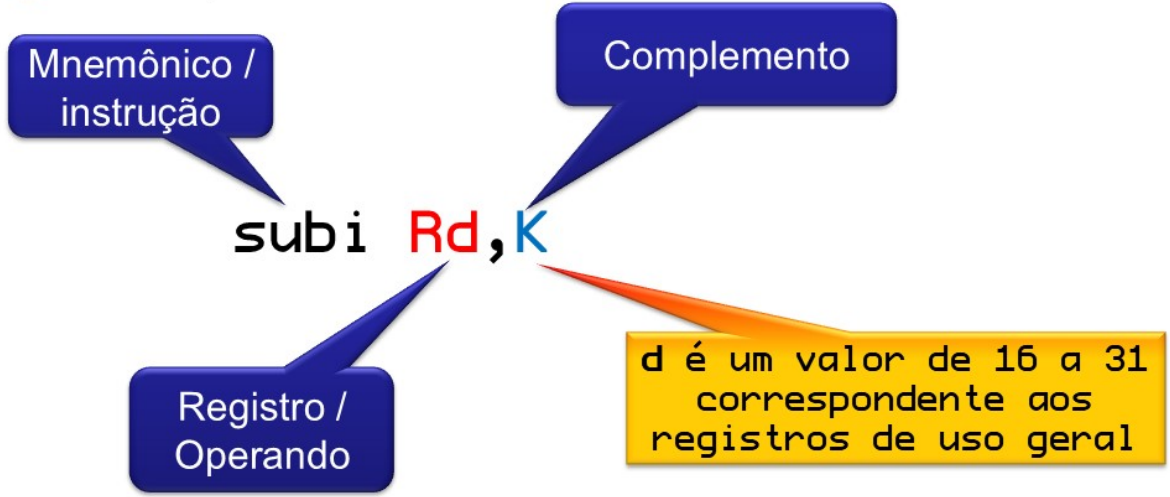
# Exemplo de operações lógicas e aritméticas

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|---|---|---|---|---|---|
| Arithmetic and Logic Instructions | | | | | |
| ADD | Rd, Rr | Add two registers | Rd ← Rd + Rr | Z,C,N,VH | 1 |
| ADC | Rd, Rr | Add with carry two registers | Rd ← Rd + Rr + C | Z,C,N,V,H | 1 |
| ADIW | Rdl, K | Add immediate to word | Rdh: Rdl ← Rdh: Rdl + K | Z,C,N,V,S | 2 |
| SUB | Rd, Rr | Subtract two registers | Rd ← Rd – Rr | Z,C,N,V,H | 1 |
| SUBI | Rd, K | Subtract constant from register | Rd ← Rd – K | Z,C,N,V,H | 1 |
| SBC | Rd, Rr | Subtract with carry two registers | Rd ← Rd – Rr – C | Z,C,N,V,H | 1 |
| SBCI | Rd, K | Subtract with carry constant from reg. | Rd ← Rd – K – C | Z,C,N,V,H | 1 |
| SBIW | Rdl, K | Subtract immediate from word | Rdh: Rdl ← Rdh: Rdl – K | Z,C,N,V,S | 2 |
| AND | Rd, Rr | Logical AND registers | Rd ← Rd × Rr | Z,N,V | 1 |
| ANDI | Rd, K | Logical AND register and constant | Rd ← Rd × K | Z,N,V | 1 |
| OR | Rd, Rr | Logical OR registers | Rd ← Rd v Rr | Z,N,V | 1 |
| ORI | Rd, K | Logical OR register and constant | Rd ← Rd v K | Z,N,V | 1 |
| EOR | Rd, Rr | Exclusive OR registers | Rd ← Rd ⊕ Rr | Z,N,V | 1 |
| COM | Rd | One's complement | Rd ← 0xFF – Rd | Z,C,N,V | 1 |
| NEG | Rd | Two's complement | Rd ← 0x00 – Rd | Z,C,N,V,H | 1 |
| SBR | Rd, K | Set bit(s) in register | Rd ← Rd v K | Z,N,V | 1 |
| CBR | Rd, K | Clear bit(s) in register | Rd ← Rd × (0xFF – K) | Z,N,V | 1 |
| INC | Rd | Increment | Rd ← Rd + 1 | Z,N,V | 1 |
| DEC | Rd | Decrement | Rd ← Rd – 1 | Z,N,V | 1 |

© Prof. Vargas

# Composição de cada comando

Exemplo:




```
subi R31,0x2E
```

© Prof. Vargas

# Do datasheet

## 124. SUBI – Subtract Immediate

### 124.1. Description

Subtracts a register and a constant, and places the result in the destination register Rd. This instruction is working on Register R16 to R31 and is very well suited for operations on the X, Y, and Z-pointers.

Operation:

(i) Rd ← Rd - K

| | Syntax: | Operands: | Program Counter: |
|---|---|---|---|
| (i) | SUBI Rd,K | 16 ≤ d ≤ 31, 0 ≤ K ≤ 255 | PC ← PC + 1 |

16-bit Opcode:

| 0101 | KKKK | dddd | KKKK |
|---|---|---|---|

```
subi R31,0x2E = 0101.0010.0001.1110
```

© Prof. Vargas

© Prof. Vargas