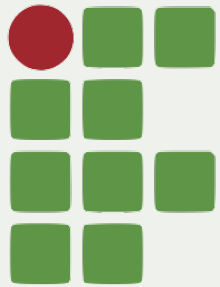
Arquitetura AVR

ATmega328p

Prof. Roberto de Matos

roberto.matos@ifsc.edu.br

09/2019



INSTITUTO FEDERAL
Santa Catarina

Câmpus São José

Objetivo

Objetivo

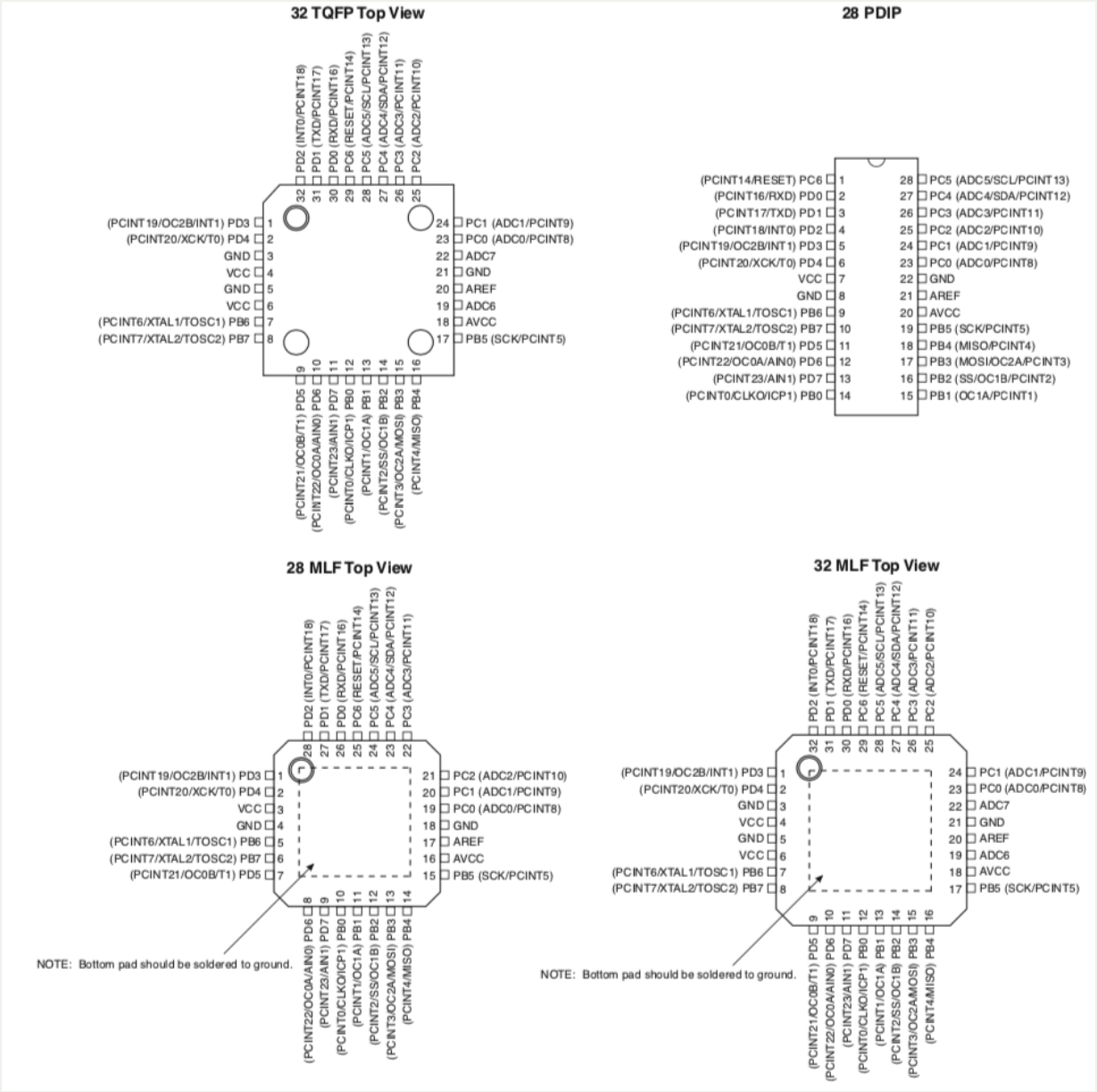
- Visão Geral dos Microcontroladores ATmega48A/PA/88A/PA/168A/PA/328/P
- Entender o Núcleo do AVR (Arquitetura)
- Entender a Hierarquia de memória

Visão Geral

Características Principais - AVR

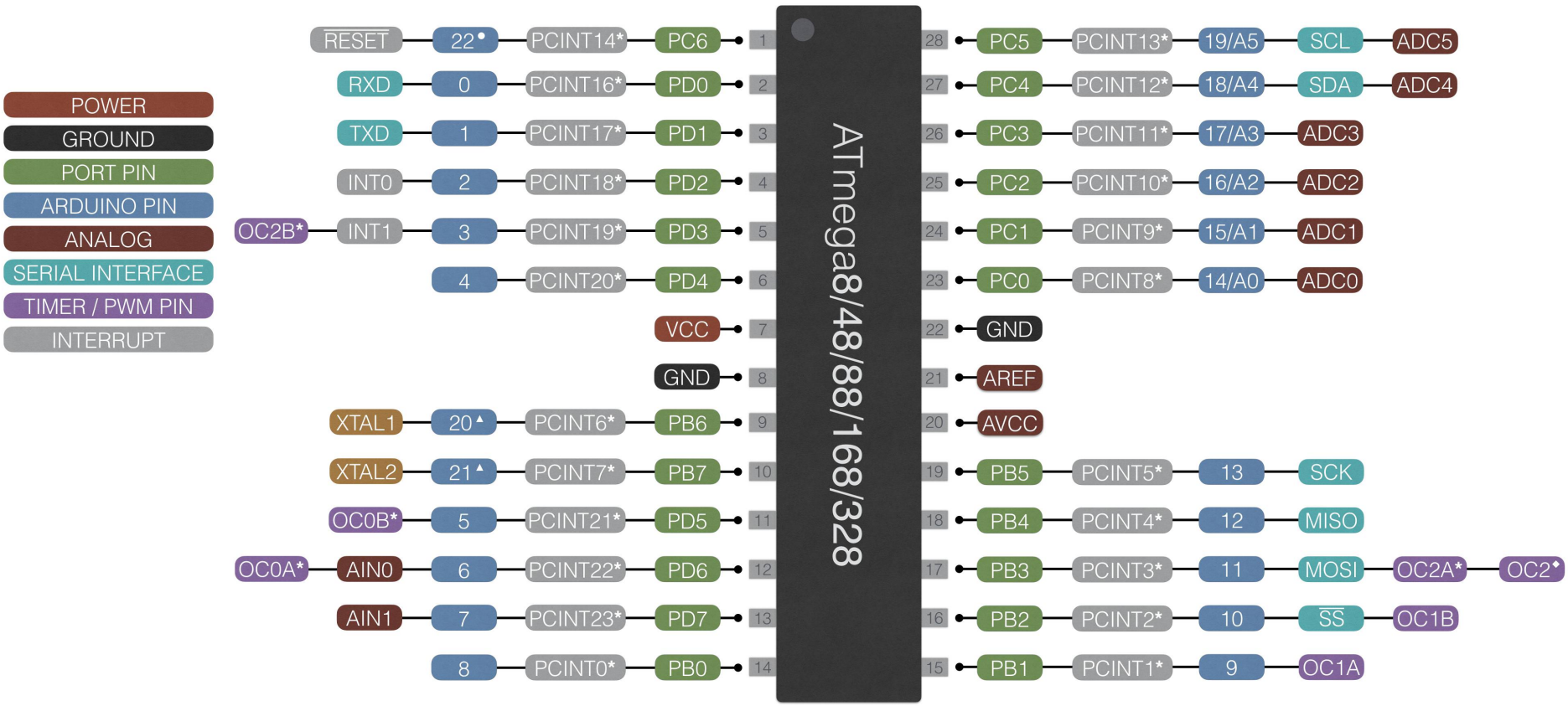
- Microcontrolador de 8 bits
- Baixo consumo
- Arquitetura RISC
- Maioria das instruções são executadas em 1 ciclo de clock

Encapsulamentos e Pinos



Pinos do Encapsulamento DIP

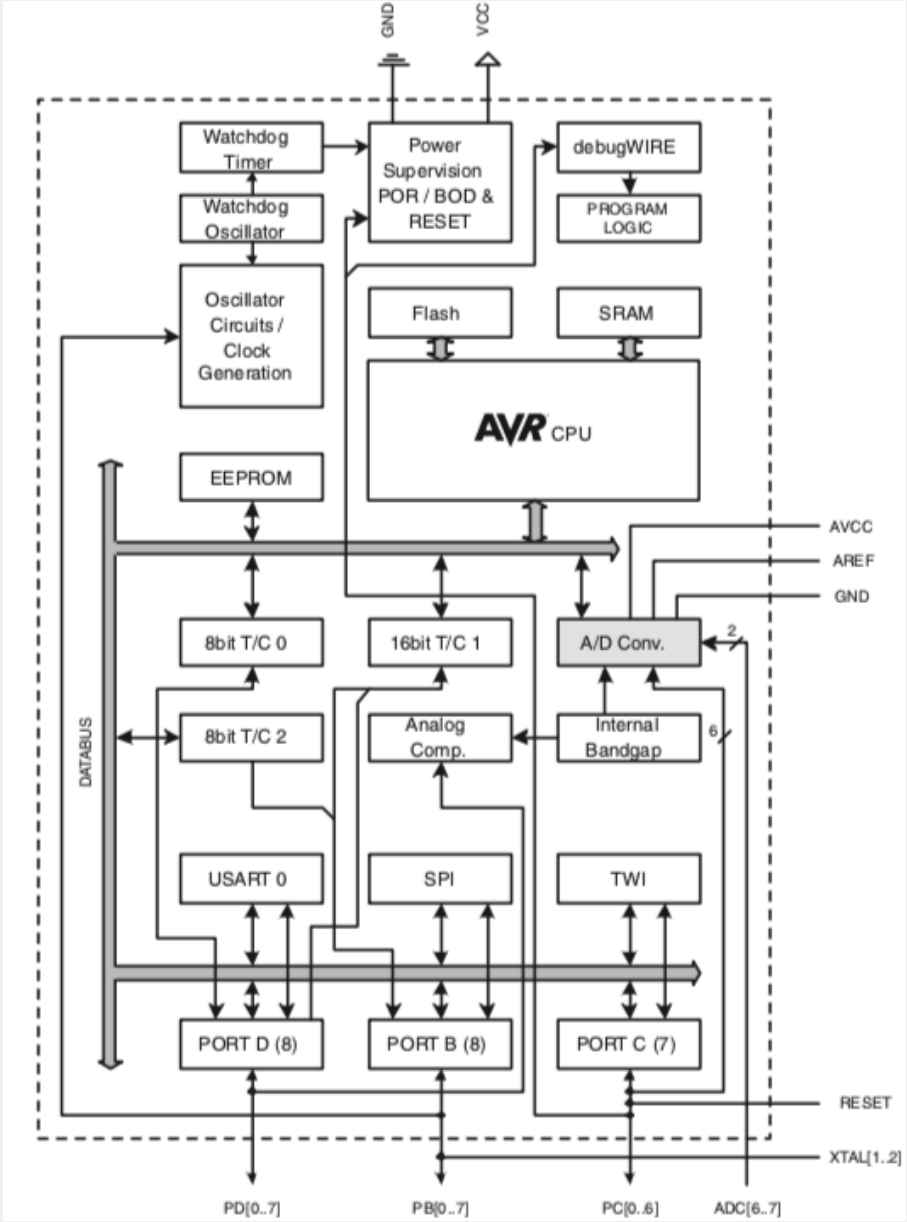
ATmega8/48/88/168/328 DIP pinout



* ATmega48/88/168/328 only
◆ ATmega8 only
▲ Internal oscillator must be enabled
• Reset pin must be disabled

http://github.com/MCUdude/MiniCore

Diagrama de Blocos do Microcontrolador AVR



Características Gerais

- 4/8/16/32KBytes de FLASH reprogramável pelo sistema
- 512/1K/1K/2Kbytes SRAM
- 256/512/512/1Kbytes EEPROM
- 23 linhas de entrada e saída de uso geral (GPIO)
- 32 registradores de uso geral
- 3 Temporizadores/Contadores com comparadores
- Interrupções internas e externas

Características Gerais

- Interface serial programável (USART)
- Interface serial de dois fios (2-wire)
- ADC de 10-bit
  - 6 canais no DIP
  - 8 canais nos TQFP e QFN/MLF)
- Watchdog programável com oscilador interno
- 5 modos de economia de energia

Diferenças ATmega48A/PA/88A/PA/168A/PA/328/

- Tamanho das memórias
- Tamanho dos Vetores de Interrupção
- Suporte do bootloader:
  - ATmega88A/PA/168A/PA/328/P suporta mecanimso de auto-programação "real" (*Read-While-Write*) com seção de *Boot Loader* separada e as instruções SPM somente podem rodar dessa seção.
  - ATmega48A/48PA não possui mecanismo *Read-While-Write* e não possui seção de *Boot Loader*. Instruções SPM podem rodar de qualquer parte do código

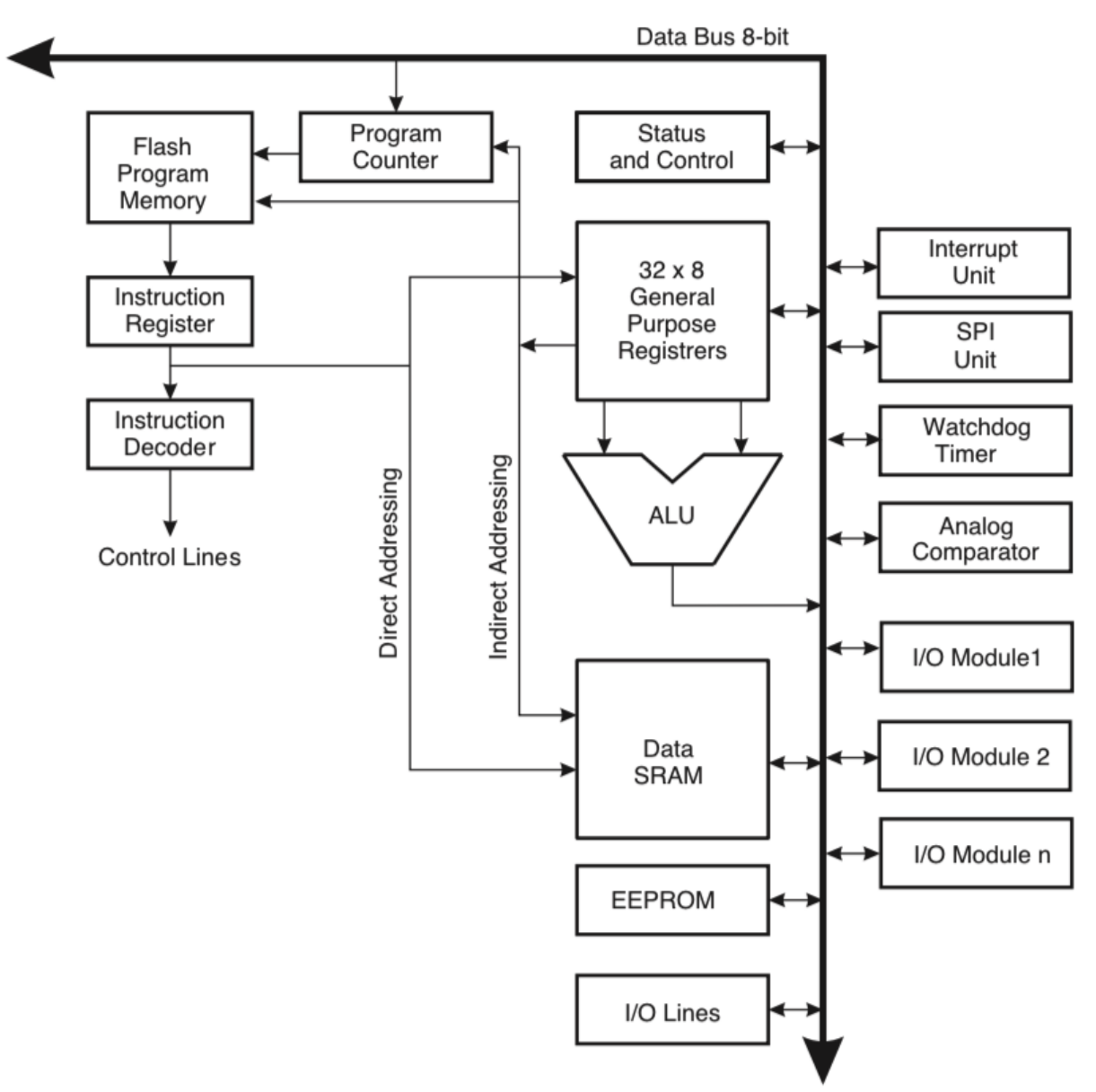
Diferenças ATmega48A/PA/88A/PA/168A/PA/328/

Tamanho das Memórias por Modelo:

Device	Flash	EEPROM	RAM
ATmega48A	4KBytes	256Bytes	512Bytes
ATmega48PA	4KBytes	256Bytes	512Bytes
ATmega88A	8KBytes	512Bytes	1KBytes
ATmega88PA	8KBytes	512Bytes	1KBytes
ATmega168A	16KBytes	512Bytes	1KBytes
ATmega168PA	16KBytes	512Bytes	1KBytes
ATmega328	32KBytes	1KBytes	2KBytes
ATmega328P	32KBytes	1KBytes	2KBytes

Núcleo do AVR

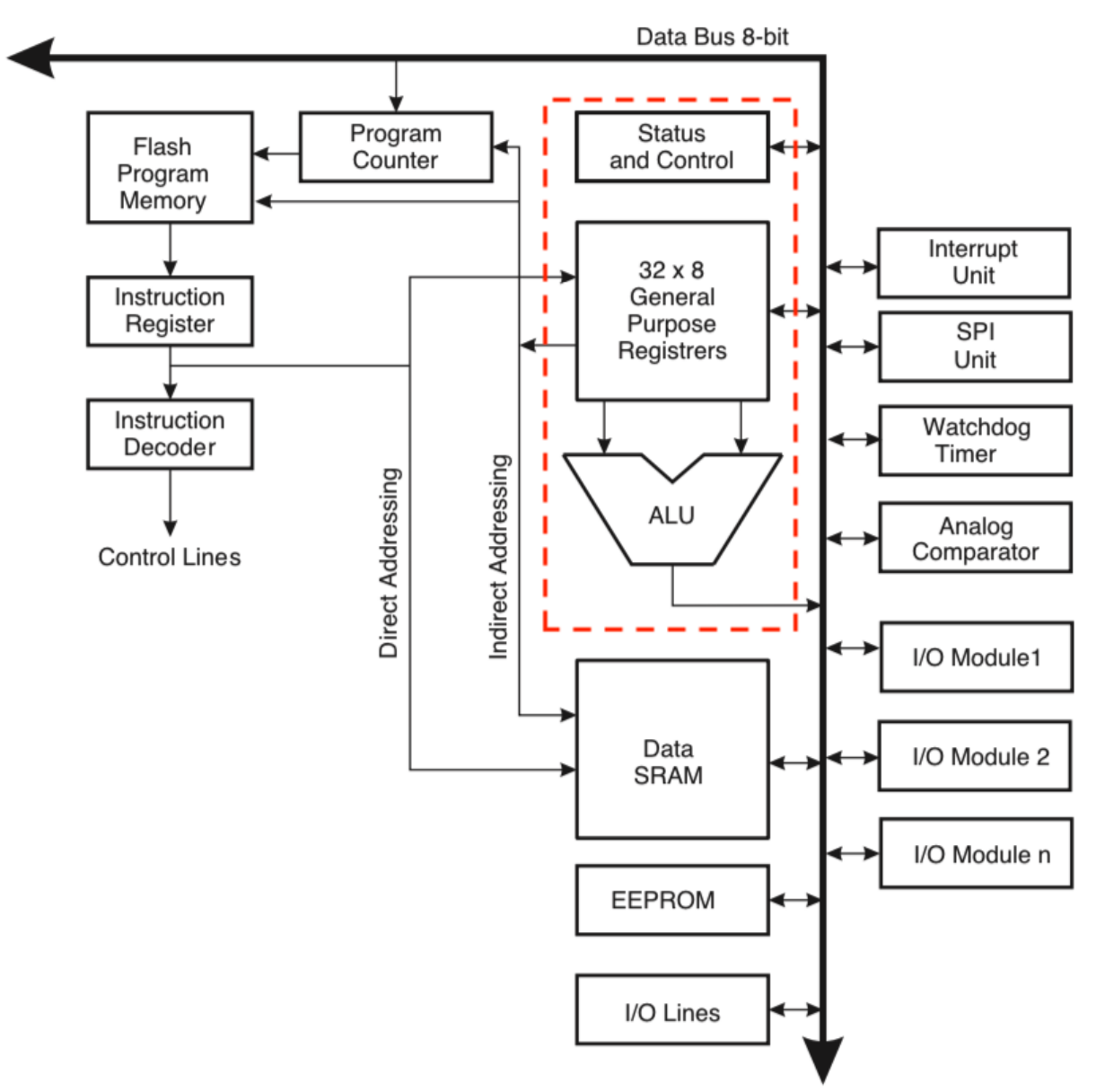
Diagrama de Blocos do Núcleo do AVR



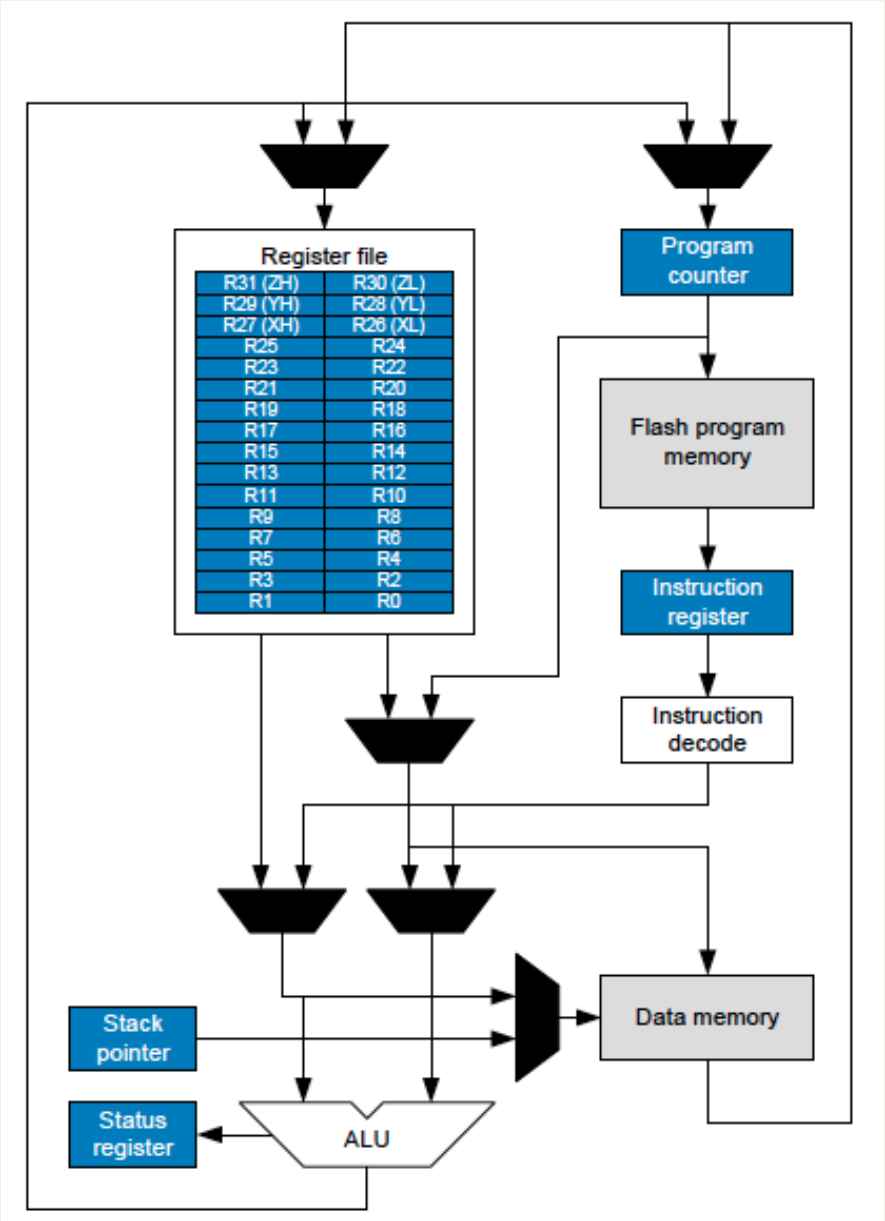
Característica AVR

- **Arquitetura Harvard modificada:** Além de possuir barramentos separados para memória de dados (SRAM) e programa (FLASH), permite ler e escrever a memória de programa usando instruções especiais.
- **Pipeline de um nível:** Enquanto uma instrução está sendo executada, a próxima instrução é buscada na memória do programa. Este conceito permite que as instruções sejam executadas em cada ciclo do relógio, ou seja, a performance de um AVR de 16 MHz é aproximadamente 16 MIPs (Milhões de Instruções por Segundo).
- A **simplicidade** da arquitetura se traduz em uma curva de aprendizado mais rápida.

Unidade Lógica Aritmética



Unidade Lógica Aritmética



Unidade Lógica Aritmética

- **Operandos:** Registradores de uso geral ou entre um registrador e um imediato (constante armazenada na memória de programa).
- **Operações:** Aritmética, lógica e funções de bit. Alguns microcontroladores da família também provêem suporte para multiplicação (com sinal, sem sinal e fracionário).
- **Status:** Após uma operação na ULA, o Registrador de Status é atualizado para refletir as informações sobre o resultado da operação.

Unidade Lógica Aritmética

- **Operandos:** Registradores de uso geral ou entre um registrador e um imediato (constante armazenada na memória de programa).
- **Operações:** Aritmética, lógica e funções de bit. Alguns microcontroladores da família também provêem suporte para multiplicação (com sinal, sem sinal e fracionário).
- **Status:** Após uma operação na ULA, o Registrador de Status é atualizado para refletir as informações sobre o resultado da operação.

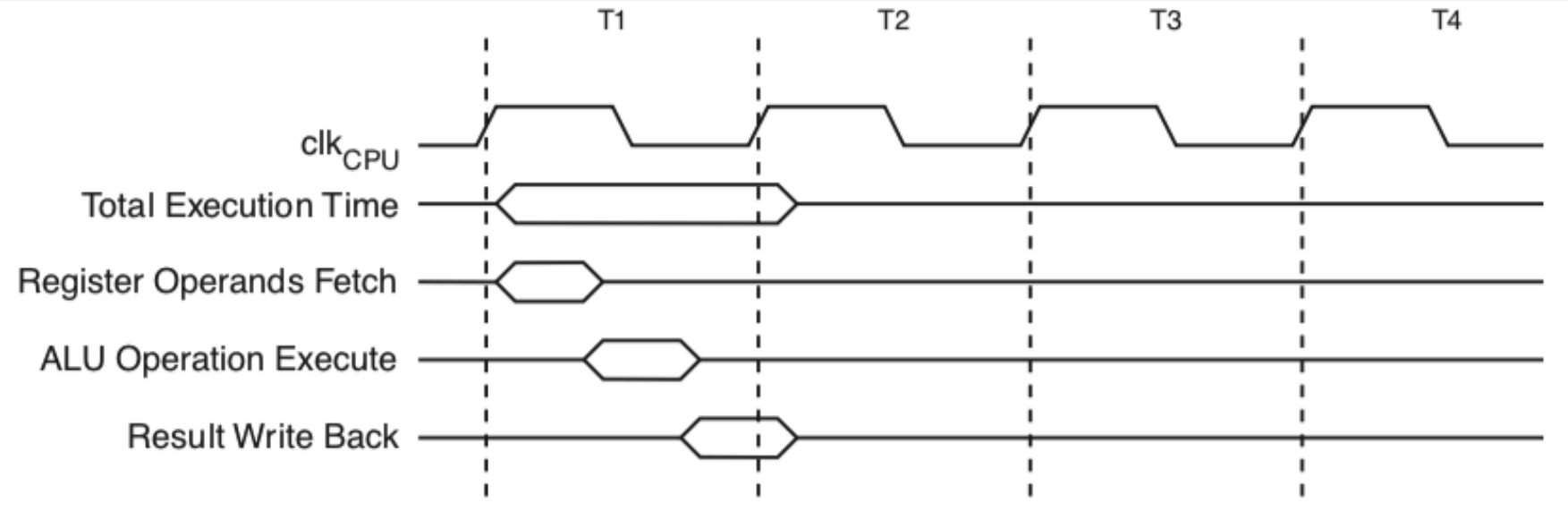
Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ULA - Exemplo Instruções

ARITHMETIC AND LOGIC INSTRUCTIONS					
Mnemonics	**Operands**	**Description**	**Operation**	**Flags**	**#Clocks**
ADD	Rd, Rr	Add two Registers without Carry	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add two Registers with Carry	Rd ← Rd + Rr + C	Z,C,N,V,H	1
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 two Registers with Carry	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract Constant from Reg with Carry.	Rd ← Rd - K - C	Z,C,N,V,H	1
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
TST	Rd	Test for Zero or Minus	Rd ← Rd · Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← 0xFF	None	1

Ciclo de Operação da ULA

Em um único ciclo de relógio a ULA executa uma operação usando dois operandos de registro e o resultado é armazenado de volta no registro de destino.

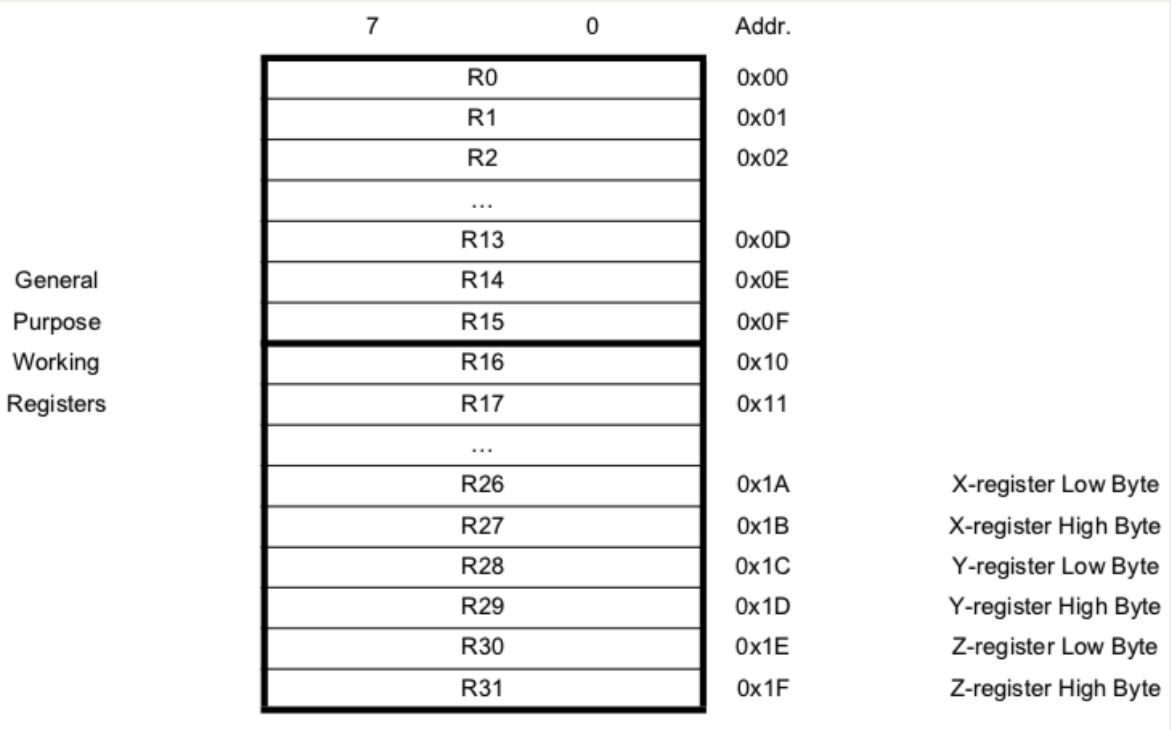


Forma de Acesso

Os registradores de uso geral são otimizados para o conjunto de instruções do AVR. Para obter o desempenho e a flexibilidade necessários, os seguintes esquemas de entrada/saída são suportados pelo banco de registradores.

Operando(s) Saída	**Operando(s) Entrada**
1x de 8 bits	1x de 8 Bits
2x de 8 bits	1x de 8 Bits
2x de 8 bits	1x de 16 Bits
1x de 16 bits	1x de 16 Bits

Registradores de Uso Geral



Mapeamento em RAM

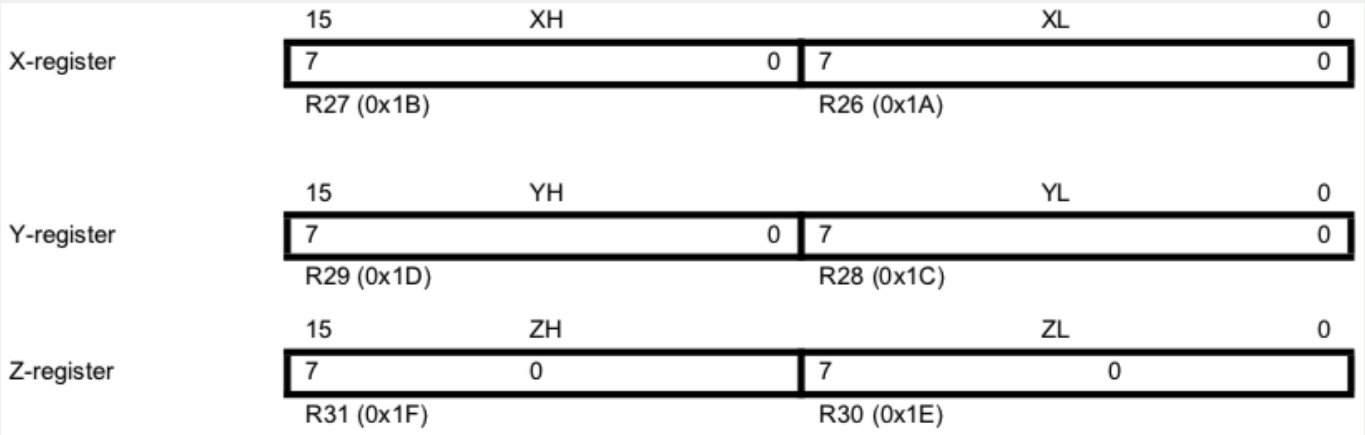
Cada registro também recebe um endereço de memória de dados. Embora não seja fisicamente implementada como localizações SRAM, essa organização de memória oferece grande flexibilidade.

Registradores X, Y e Z

- Os registradores R26 ... R31 também possuem funções especiais. Eles são ponteiros de endereço de 16 bits para endereçamento indireto, definidos como X, Y e Z.
- Nos diferentes modos de endereçamento, esses registradores de endereços têm funções como deslocamento fixo, incremento automático e decréscimo automático.

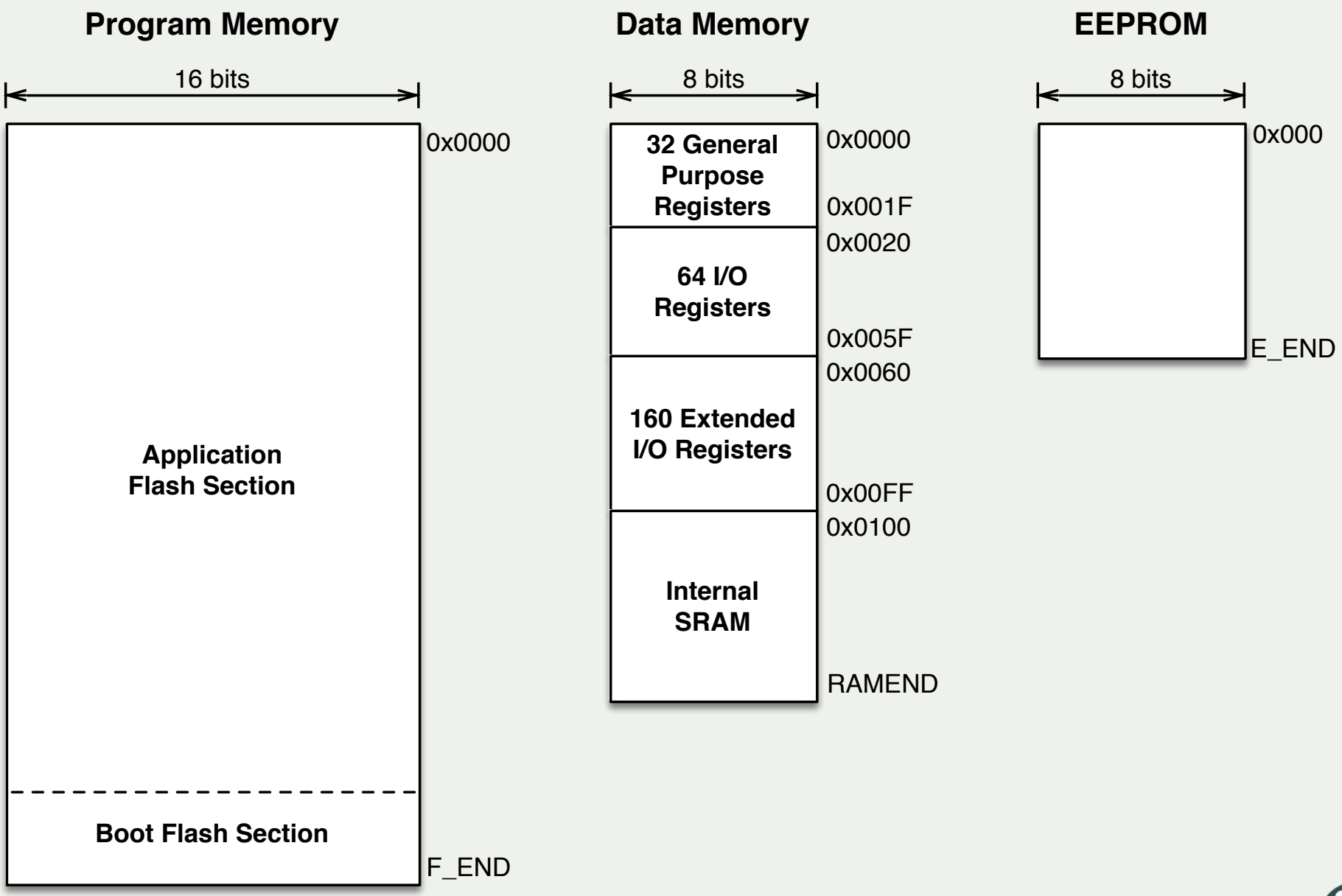
Registradores X, Y e Z

- Os registradores R26 ... R31 também possuem funções especiais. Eles são ponteiros de endereço de 16 bits para endereçamento indireto, definidos como X, Y e Z.
- Nos diferentes modos de endereçamento, esses registradores de endereços têm funções como deslocamento fixo, incremento automático e decréscimo automático.



Memórias do AVR

Memórias – Organização



Memórias - Modelos vs. Tamanho

- Endereços Finais vs. Modelos:

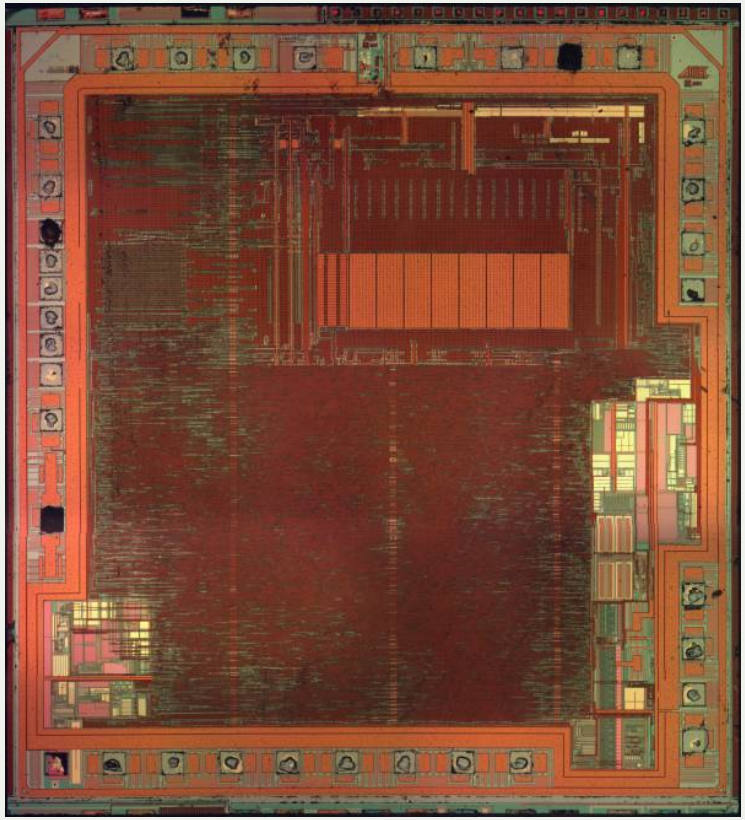
Modelo	F_END	RAMEND	E_END
ATmega88*	0x0FFF	0x04FF	0x1FF
ATmega168*	0x1FFF	0x04FF	0x1FF
ATmega328*	0x3FFF	0x08FF	0x3FF

- Tamanhos vs. Modelos:

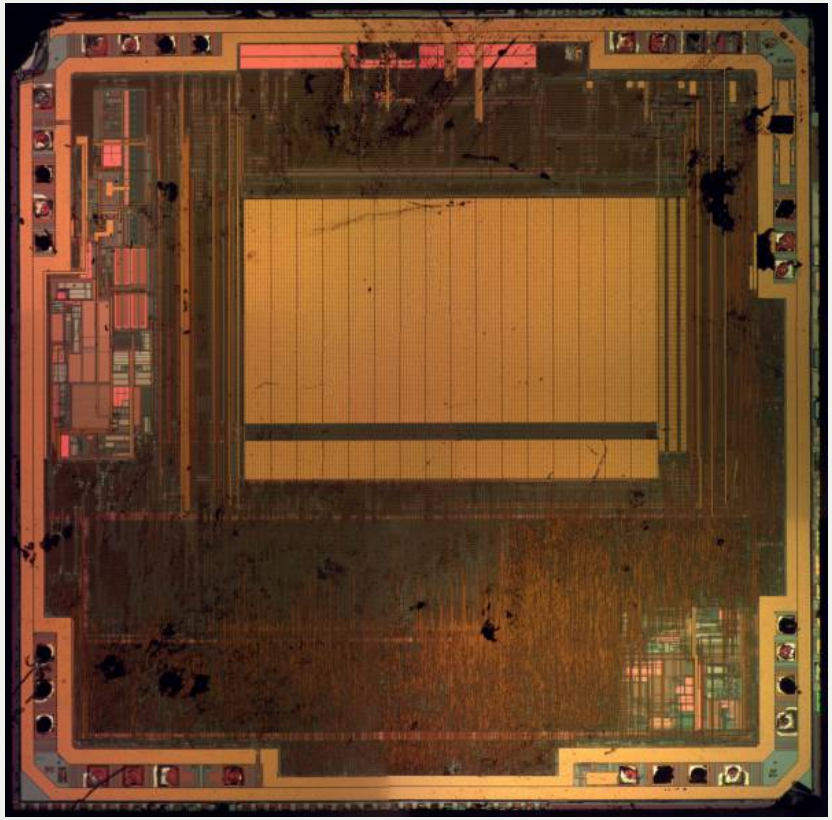
Modelo	FLASH	RAM	EEPROM
ATmega88*	4K x 16 bits	1K x 8 bits	512 x 8 bits
ATmega168*	8K x 16 bits	1K x 8 bits	512 x 8 bits
ATmega328*	16K x 16 bits	2K x 8 bits	1K x 8 bits

Memórias - Die[1]

- ATmega48: $7.01mm^2$

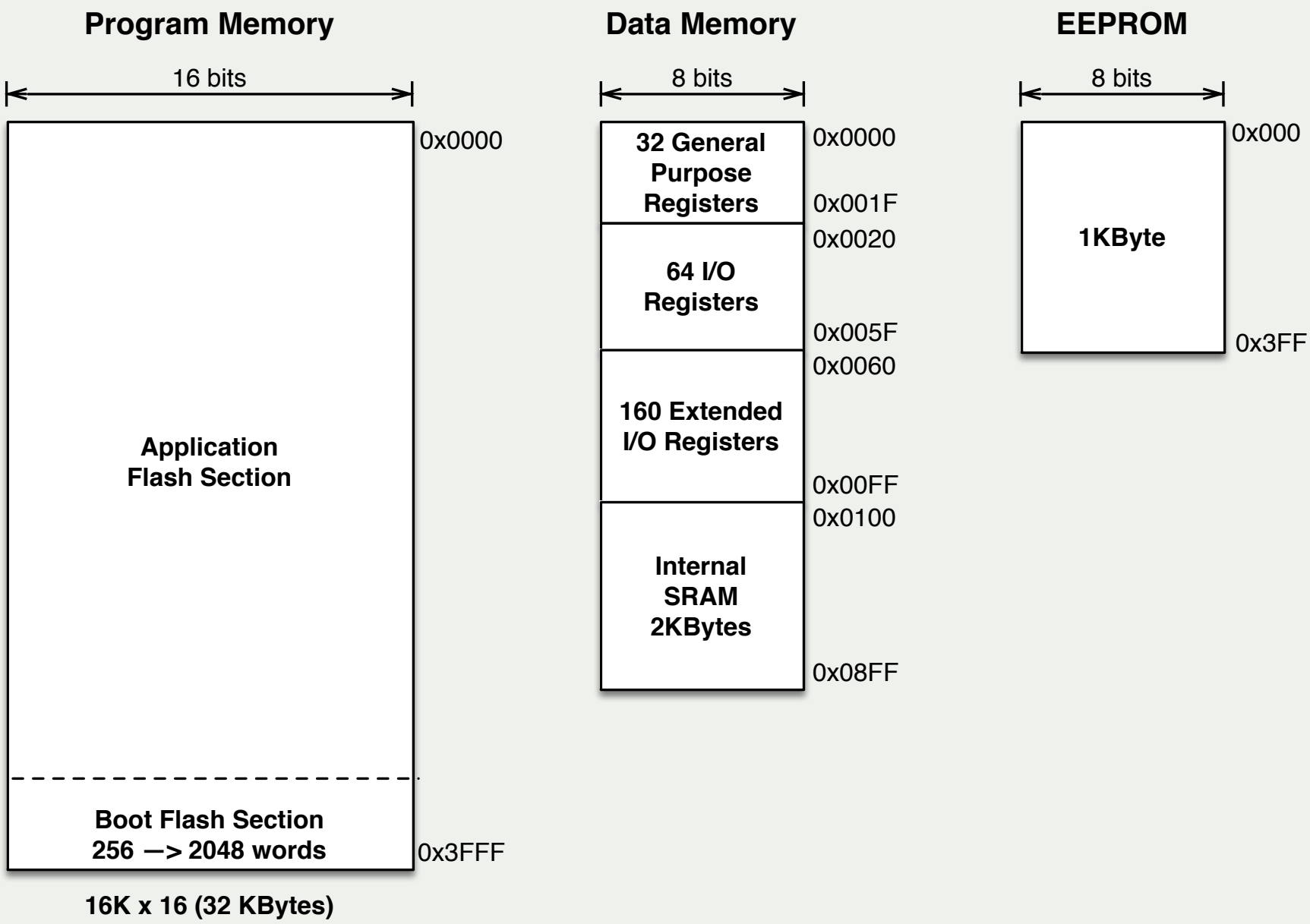


- ATmega328p: $8.82mm^2$

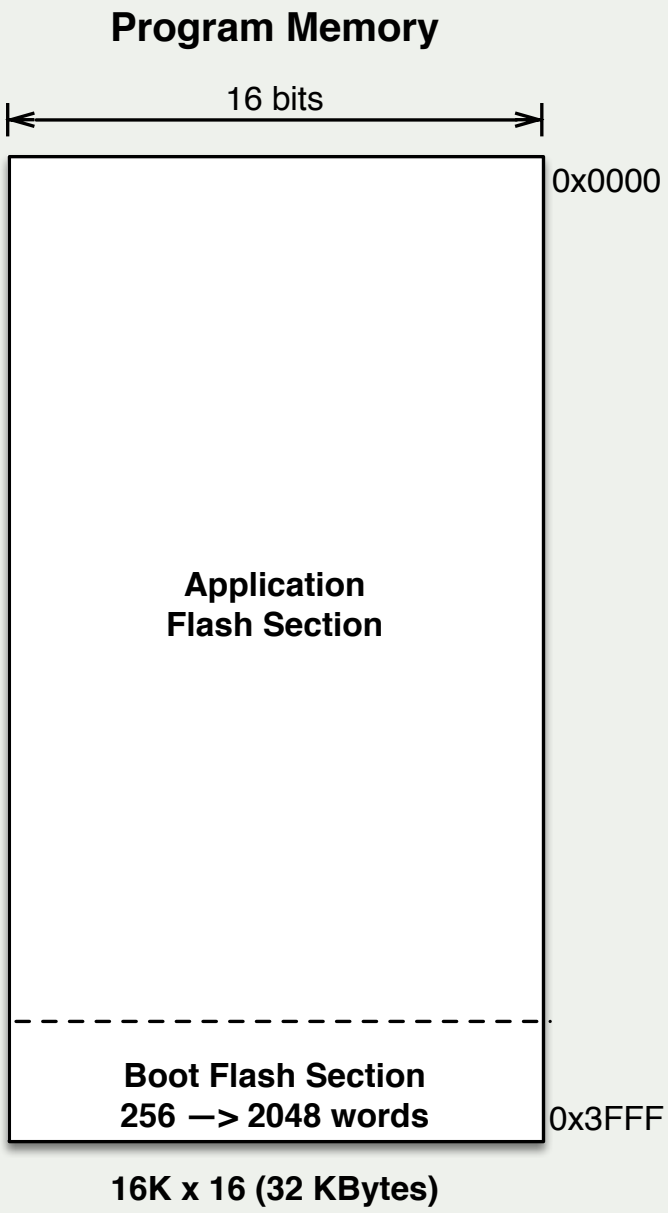


[1] Fonte: https://siliconpr0n.org

Memórias ATmega328p



Memória de Programa - FLASH

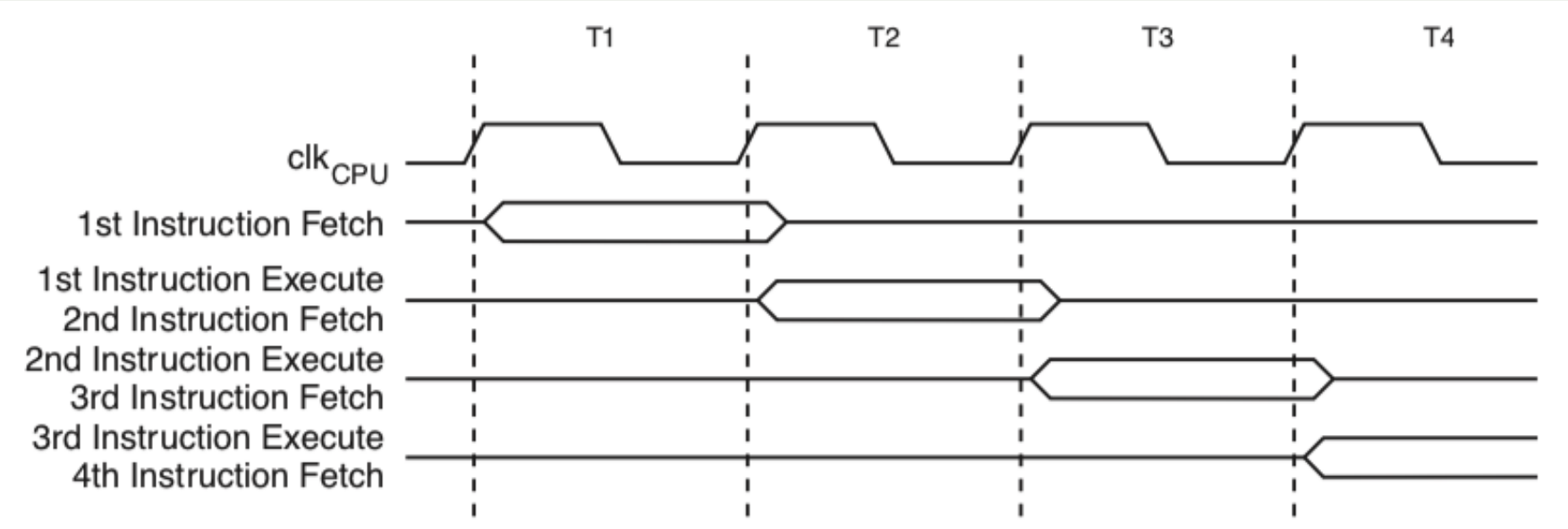


Memória de Programa - FLASH

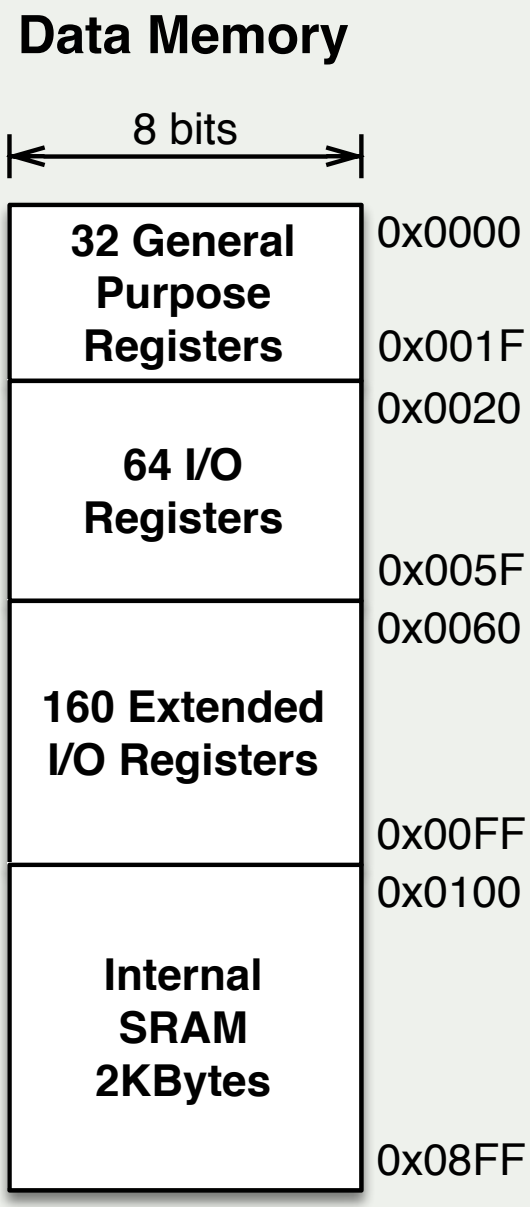
- Como todas as instruções do AVR são de 16 ou 32 bits, a palavra de memória da FLASH foi projetada para ter 16 bits, facilitando a *busca* das instruções.
- A memória tem uma durabilidade de 10.000 ciclos de re-escrita.
- O tamanho do PC para o ATmega48A/PA/88A/PA/168A/PA/328/P é de 11/12/13/14 bits respectivamente.
- Exceto pelo ATmega48*, todo os outros modelos possuem área de bootloader e permitem escrita na FLASH durante a leitura.
- Tabelas de constantes podem ser alocadas em qualquer ponto da FLASH. Para leitura é usado a instrução LPM (*Load Program Memory*).

Diagrama de tempo de Busca e Execução

O funcionamento do *pipeline* é possibilitado pela palavra de 16 bits da memória de programa, a arquitetura Harvard e o conceito de acesso rápido ao banco de registradores.

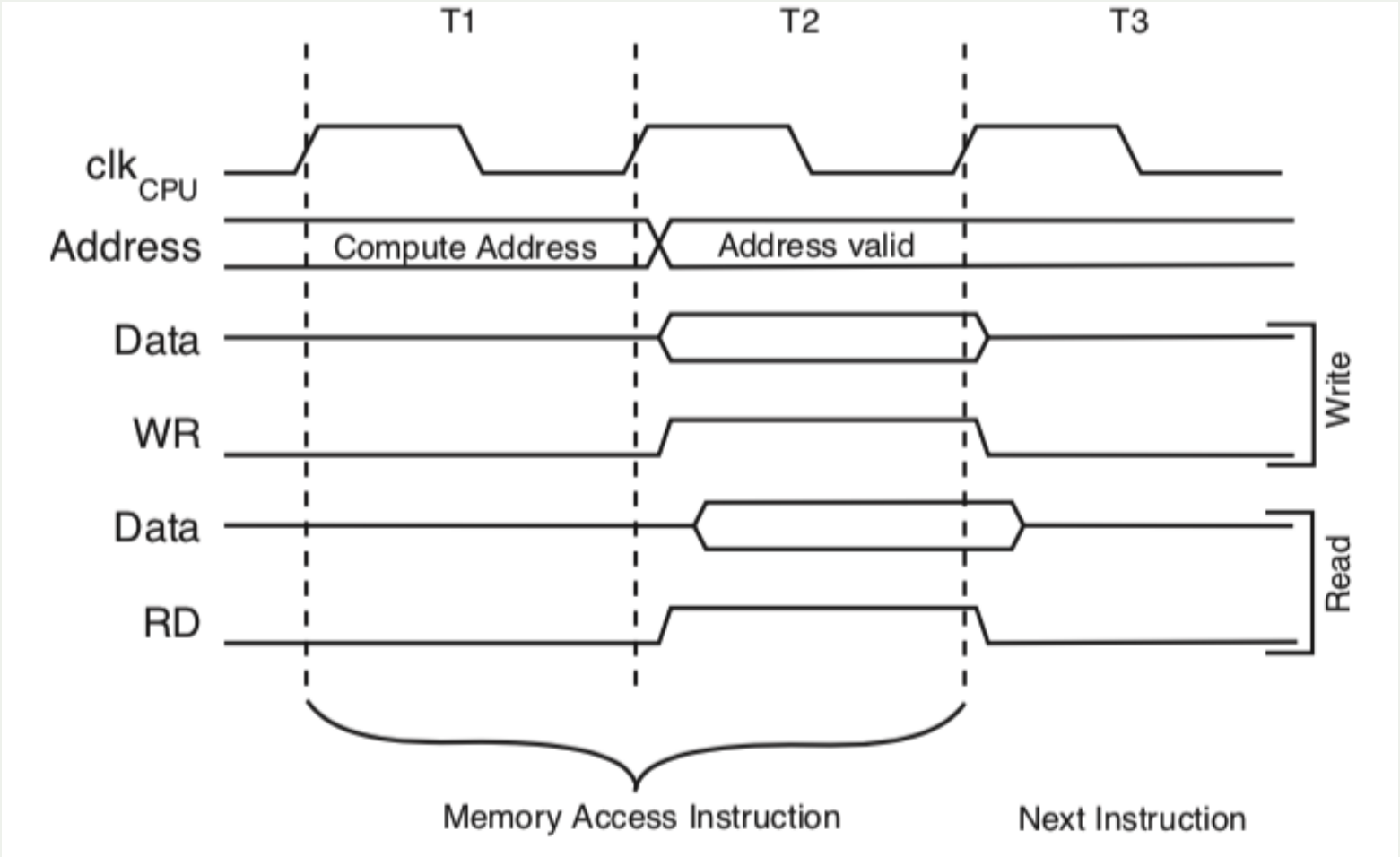


Memória de Dados – Regiões



Memória de Dados – Tempos de Acesso

O acesso aos dados da SRAM é realizado em dois ciclos de relógio.



Espaço de Endereçamento de I/O vs. I/O mapeado em memória

- As portas I/O são tradicionalmente tratadas como partes separadas de um computador.
- O AVR inclui as instruções *IN* e *OUT* para acesso ao I/O, representado pelos 64 registradores da região de I/O.

Espaço de Endereçamento de I/O vs. I/O mapeado em memória

- As portas I/O são tradicionalmente tratadas como partes separadas de um computador.
- O AVR inclui as instruções *IN* e *OUT* para acesso ao I/O, representado pelos 64 registradores da região de I/O.

Problema

A família ATmega precisa de mais de 64 registradores de I/O devido a quantidade de periféricos (GPIO, Timers, ADCs, ...)

Espaço de Endereçamento de I/O vs. I/O mapeado em memória

- As portas I/O são tradicionalmente tratadas como partes separadas de um computador.
- O AVR inclui as instruções *IN* e *OUT* para acesso ao I/O, representado pelos 64 registradores da região de I/O.

Problema

A família ATmega precisa de mais de 64 registradores de I/O devido a quantidade de periféricos (GPIO, Timers, ADCs, ...)

Solução

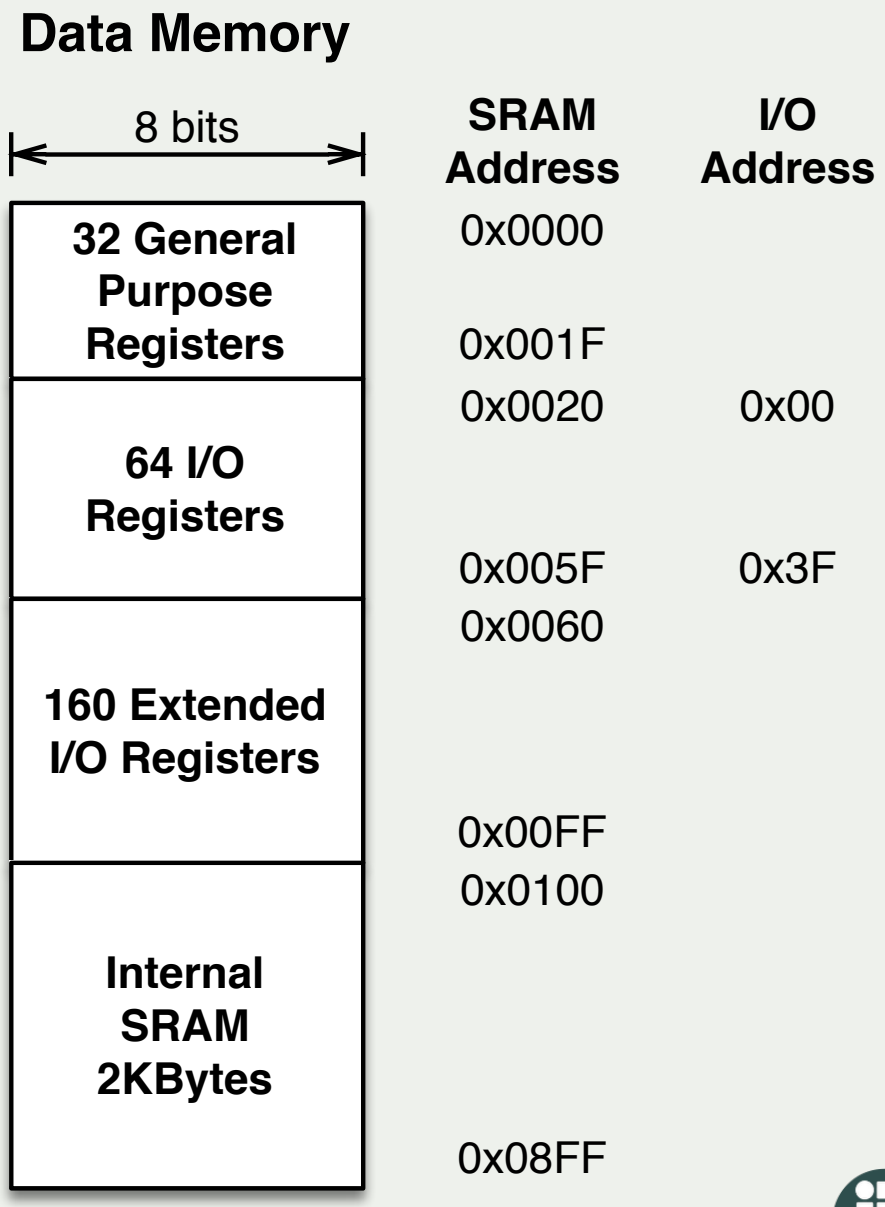
Mudança de paradigma: Em vez de 5 elementos básicos (entrada, saída, ULA, CPU, memória), simplifica-se o computador para apenas três (CPU, ULA e memória). A CPU agora pode acessar os 160 registradores de I/O "estendidos" usando instruções para acesso à SRAM (ex.: *LDS* e *STS*).

Endereçamento de I/O vs. I/O mapeado em memória

- Essa é uma técnica tão poderosa que a Atmel estendeu o mapeamento de I/O em memória para incluir os 32 registros de uso geral, os 64 registros de I/O originais e os 160 registros de I/O estendidos. A sobreposição do espaço de endereço de I/O com o espaço de endereço da SRAM é mostrada no próximo slide.
- Um benefício colateral do mapeamento duplo é o grande número de maneiras de acessar dados dentro da SRAM (modos de endereçamento) versus o número limitado de instruções e modos de endereçamento disponíveis para acessar os 64 registros de I/O originais, ou seja, *IN* e *OUT*.
- É muito importante perceber que os registros não são implementados com SRAM. O mapeamento é simplesmente uma maneira conveniente de acessar os registradores localizados fisicamente em diversos locais dentro do chip.

Endereçamento de I/O vs. I/O mapeado em memória

- Instruções *IN*/*OUT* acessam os 64 registradores de I/O.
- Instruções *ST*/*STS*/*STD* and *LD*/*LDS*/*LDD* acessam toda a memória de dados



Primeiro Programa Assembly

- Abra a Wiki da Disciplina ...