Rudi: Calculadora Rudimentar

Projetando um caminho de dados

Prof. Roberto de Matos

roberto.matos@ifsc.edu.br

INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

Objetivo

Objetivo

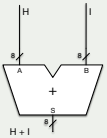
- Construir circuitos utilizando blocos de construção digital.
- Introdução ao caminho de dados e à unidade de controle.

Implementação paralela

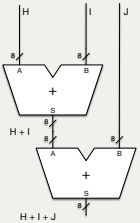
Implementação paralela

- Implementar um hardware que faça a seguinte operação:
  $\mathtt{Y} = \mathtt{H} + \mathtt{I} + \mathtt{J} + \mathtt{K} + \mathtt{L} - \mathtt{N} - \mathtt{O}$
- Considere todas as variáveis de 8 bits.

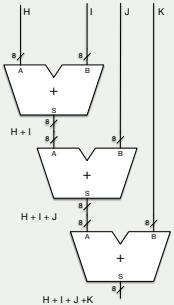
Implementando: $\mathtt{Y} = \mathtt{H} + \mathtt{I} + \mathtt{J} + \mathtt{K} + \mathtt{L} - \mathtt{N} - \mathtt{O}$



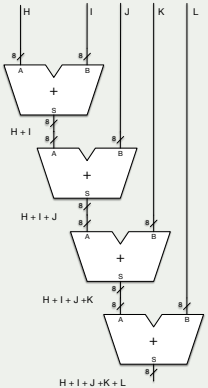
Implementando: $Y = H + I + J + K + L - N - O$



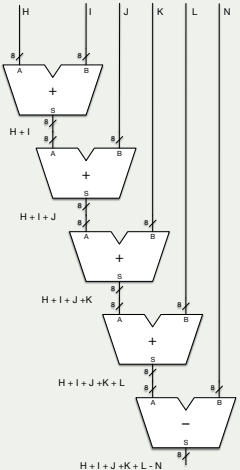
Implementando: $Y = H + I + J + K + L - N - O$



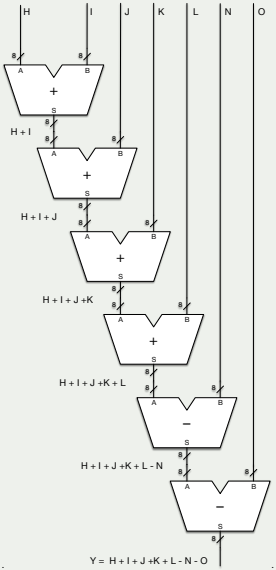
Implementando: $Y = H + I + J + K + L - N - O$



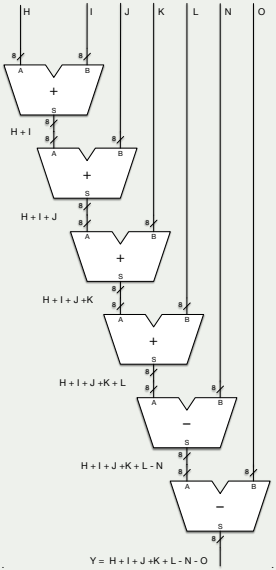
Implementando: $Y = H + I + J + K + L - N - O$



Implementando: $Y = H + I + J + K + L - N - O$



Implementando: $Y = H + I + J + K + L - N - O$



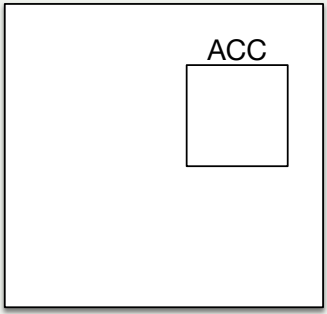
- Rápido, mas grande consumo de lógica.
- Modificação da expressão implica em modificação do hardware.
- Entrada de todos os operandos de uma vez.

Como melhorar?

- Reutilizar ao máximo o hardware.
- Executar uma operação por vez.
- Exemplo: Calculadora simples.

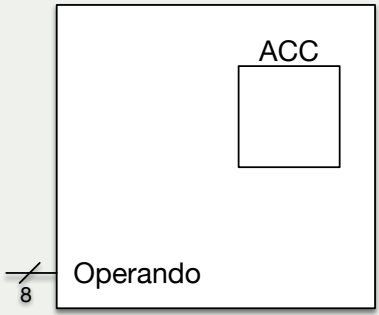
Como melhorar?

- Reutilizar ao máximo o hardware.
- Executar uma operação por vez.
- Exemplo: Calculadora simples.
- Comandos:



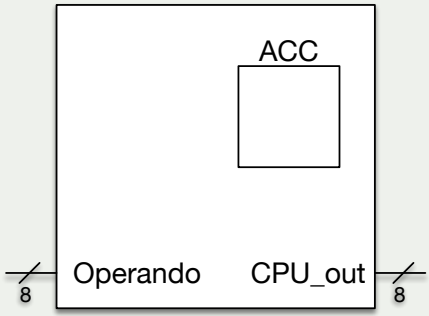
Como melhorar?

- Reutilizar ao máximo o hardware.
- Executar uma operação por vez.
- Exemplo: Calculadora simples.
- Comandos:
  - $ACC \leftarrow Operando$



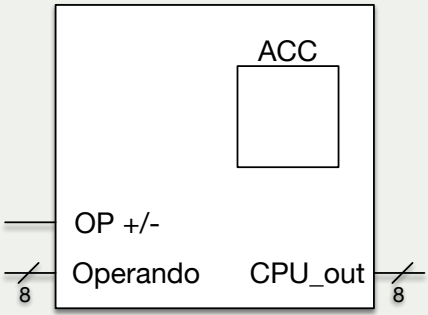
Como melhorar?

- Reutilizar ao máximo o hardware.
- Executar uma operação por vez.
- Exemplo: Calculadora simples.
- Comandos:
  - $ACC \leftarrow Operando$
  - $CPU\_out \leftarrow ACC$



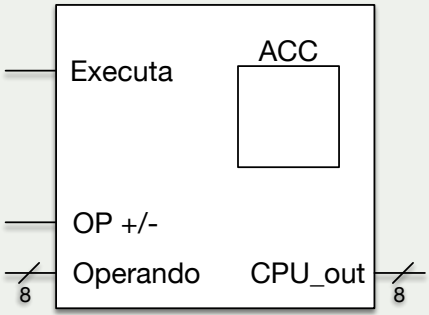
Como melhorar?

- Reutilizar ao máximo o hardware.
- Executar uma operação por vez.
- Exemplo: Calculadora simples.
- Comandos:
  - $ACC \leftarrow Operando$
  - $CPU\_out \leftarrow ACC$
  - $ACC \leftarrow ACC + Operando$
  - $ACC \leftarrow ACC - Operando$



Como melhorar?

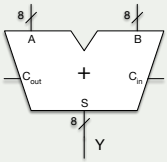
- Reutilizar ao máximo o hardware.
- Executar uma operação por vez.
- Exemplo: Calculadora simples.
- Comandos:
  - $ACC \leftarrow Operando$
  - $CPU\_out \leftarrow ACC$
  - $ACC \leftarrow ACC + Operando$
  - $ACC \leftarrow ACC - Operando$



Rudi: Caminho de dados e sinais de controle

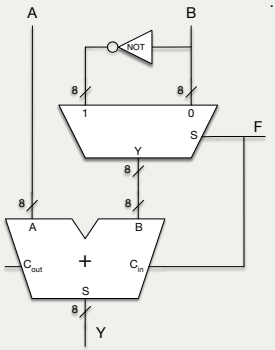
Unidade Aritmética

- Apenas um somador para soma e subtração:



Unidade Aritmética

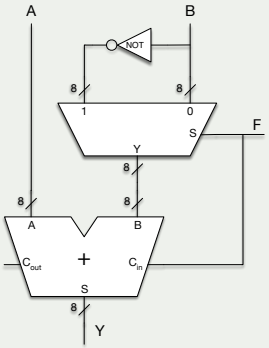
- Apenas um somador para soma e subtração:



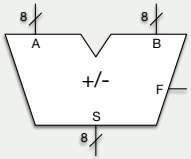
- $F = 0 \rightarrow Y = A + B$
- $F = 1 \rightarrow Y = A + \overline{B} + 1 = A - B$

Unidade Aritmética

- Apenas um somador para soma e subtração:

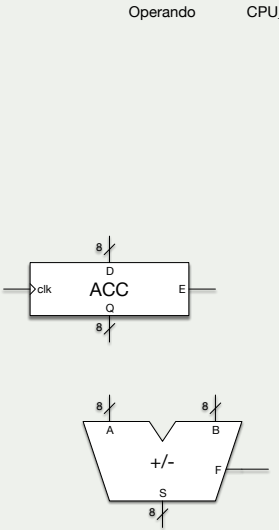


- Símbolo:



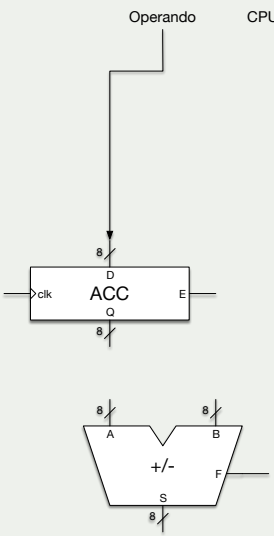
- $F = 0 \rightarrow Y = A + B$
- $F = 1 \rightarrow Y = A + \overline{B} + 1 = A - B$

Caminho dos dados



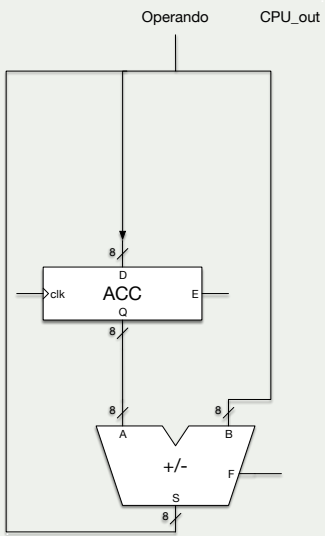
- $ACC \leftarrow Operando$
- $ACC \leftarrow ACC + Operando$
- $ACC \leftarrow ACC - Operando$
- $CPU\_out \leftarrow ACC$

Caminho dos dados



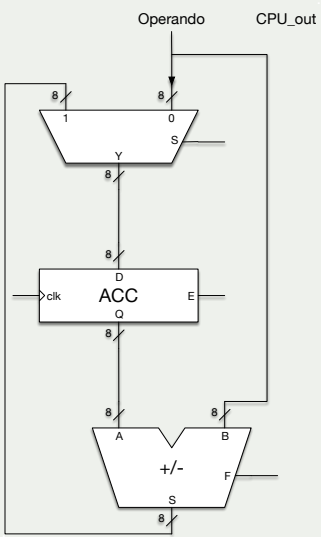
- $ACC \leftarrow Operando$
- $ACC \leftarrow ACC + Operando$
- $ACC \leftarrow ACC - Operando$
- $CPU\_out \leftarrow ACC$

Caminho dos dados



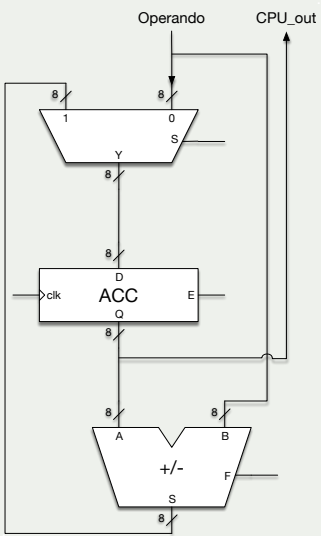
- $ACC \leftarrow Operando$
- $ACC \leftarrow ACC + Operando$
- $ACC \leftarrow ACC - Operando$
- $CPU\_out \leftarrow ACC$

Caminho dos dados



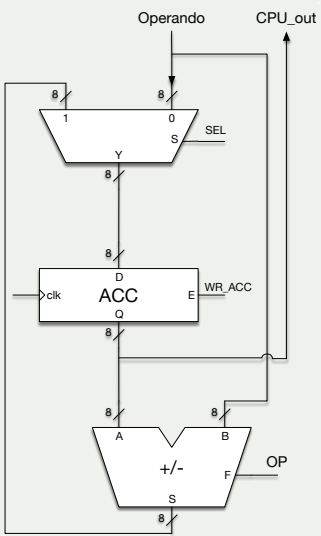
- $ACC \leftarrow Operando$
- $ACC \leftarrow ACC + Operando$
- $ACC \leftarrow ACC - Operando$
- $CPU\_out \leftarrow ACC$

Caminho dos dados



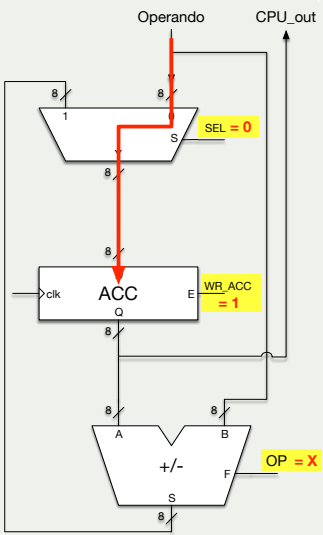
- $ACC \leftarrow Operando$
- $ACC \leftarrow ACC + Operando$
- $ACC \leftarrow ACC - Operando$
- $CPU\_out \leftarrow ACC$

Caminho dos dados



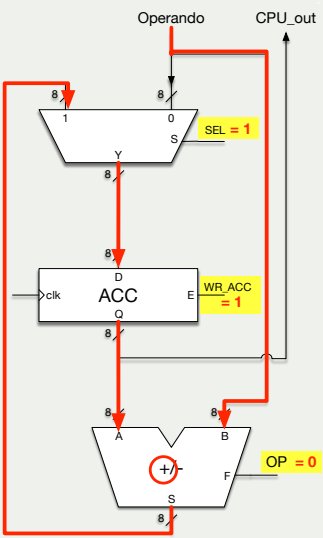
- Uma entrada: `Operando`
- Uma saída: `CPU_out`
- Três sinais de controle internos:
  - `SEL`, `WR_ACC` e `OP`

Caminho dos dados



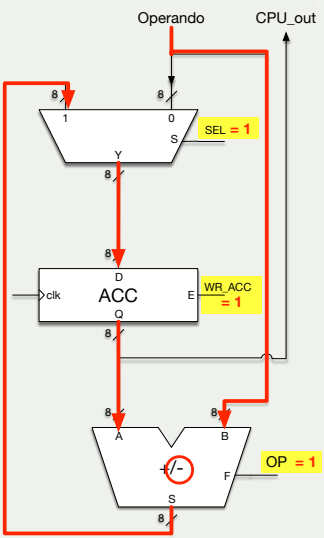
- *ACC* ← *Operando*

Caminho dos dados



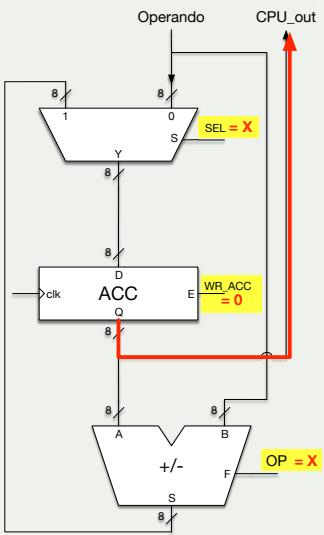
- $ACC \leftarrow ACC + Operando$

Caminho dos dados



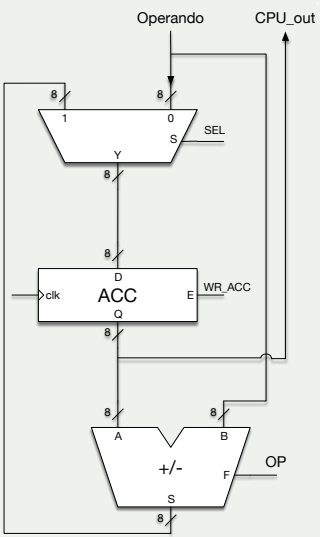
- $ACC \leftarrow ACC - Operando$

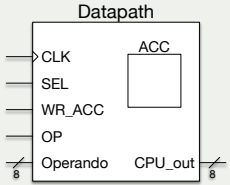
Caminho dos dados



- $CPU\_out \leftarrow ACC$

Caminho dos dados



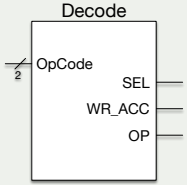


Sinais de controle e decodificador

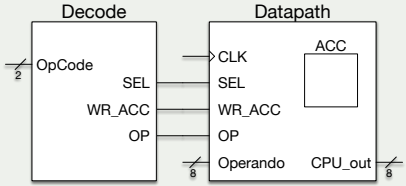
Operação	OpCode	SEL	WR_ACC	OP
$CPU\_out \leftarrow ACC$	00	X	0	X
$ACC \leftarrow Operando$	01	0	1	X
$ACC \leftarrow ACC + Operando$	10	1	1	0
$ACC \leftarrow ACC - Operando$	11	1	1	1

Sinais de controle e decodificador

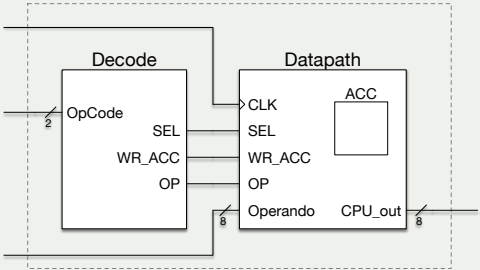
Operação	OpCode	SEL	WR_ACC	OP
$CPU\_out \leftarrow ACC$	00	X	0	X
$ACC \leftarrow Operando$	01	0	1	X
$ACC \leftarrow ACC + Operando$	10	1	1	0
$ACC \leftarrow ACC - Operando$	11	1	1	1



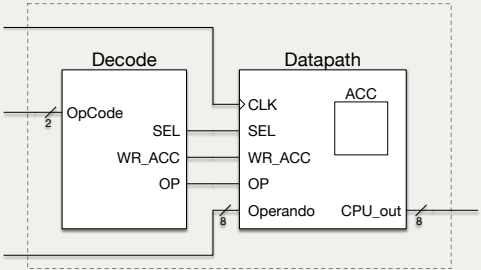
Caminho de dados e decodificador

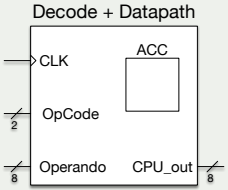


Caminho de dados e decodificador



Caminho de dados e decodificador





Calculando com o Rudi

- $\texttt{H} + \texttt{I} + \texttt{J} + \texttt{K} + \texttt{L} - \texttt{N} - \texttt{O}$
- $5 + 3 + 7 + 8 + 16 - 2 - 1$

Calculando com o Rudi

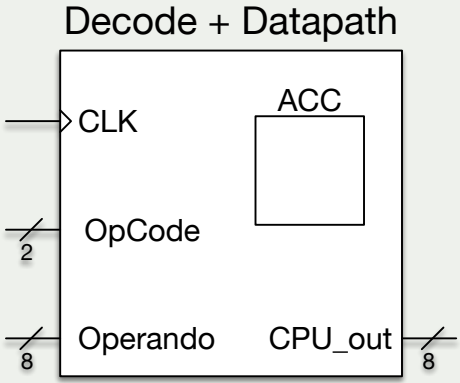
- $\texttt{H} + \texttt{I} + \texttt{J} + \texttt{K} + \texttt{L} - \texttt{N} - \texttt{O}$
- $5 + 3 + 7 + 8 + 16 - 2 - 1$

Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX

Calculando com o Rudi

- $\texttt{H} + \texttt{I} + \texttt{J} + \texttt{K} + \texttt{L} - \texttt{N} - \texttt{O}$
- $5 + 3 + 7 + 8 + 16 - 2 - 1$

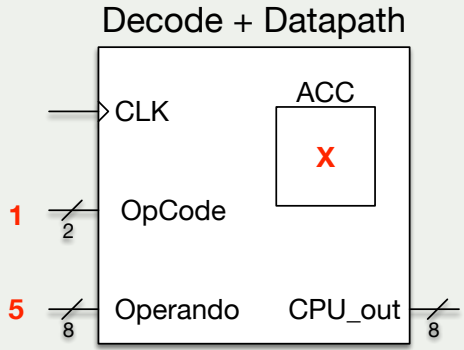
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

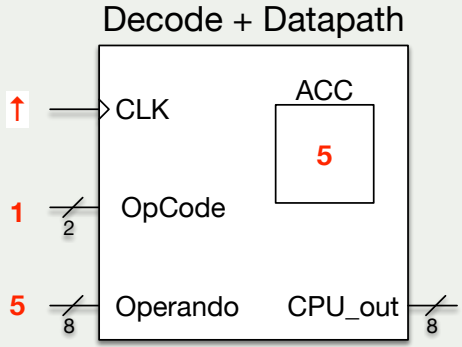
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

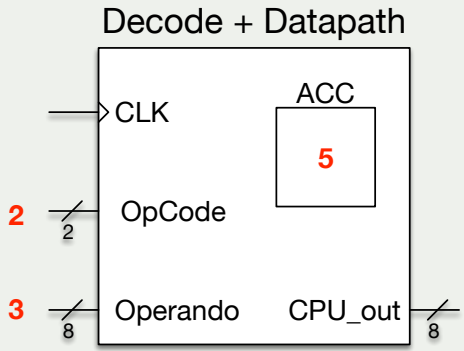
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

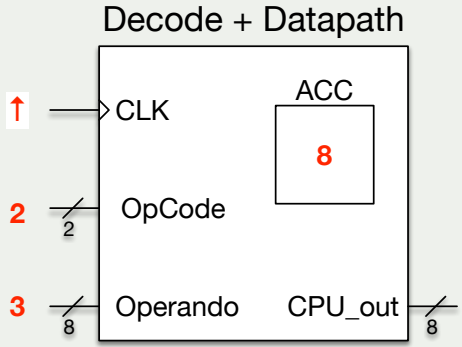
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

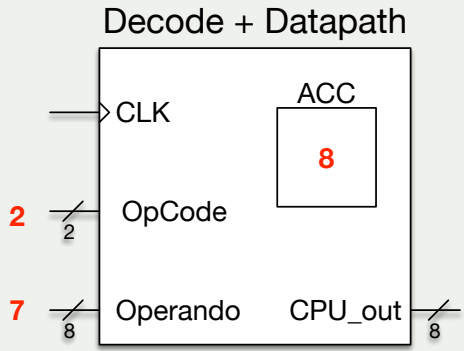
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

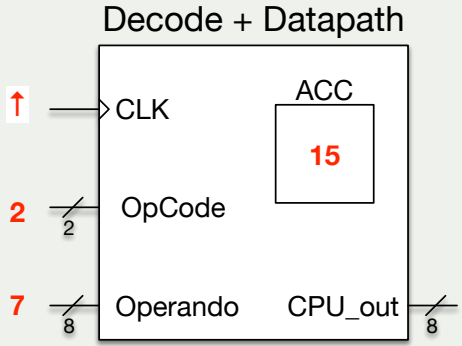
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- $\texttt{H} + \texttt{I} + \texttt{J} + \texttt{K} + \texttt{L} - \texttt{N} - \texttt{O}$
- $5 + 3 + 7 + 8 + 16 - 2 - 1$

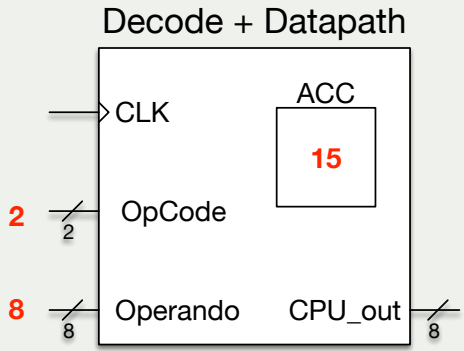
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

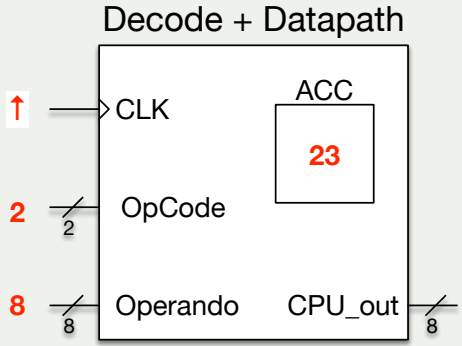
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

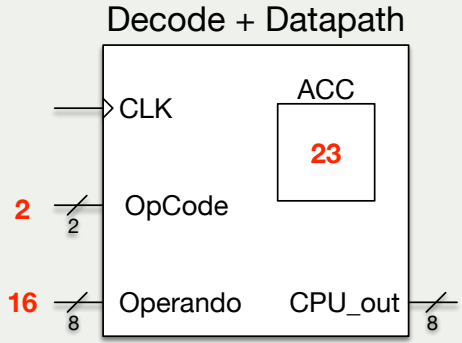
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

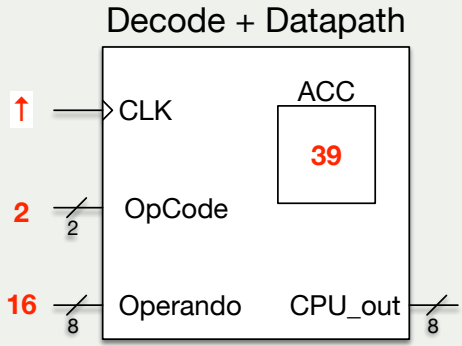
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- $\texttt{H} + \texttt{I} + \texttt{J} + \texttt{K} + \texttt{L} - \texttt{N} - \texttt{O}$
- $5 + 3 + 7 + 8 + 16 - 2 - 1$

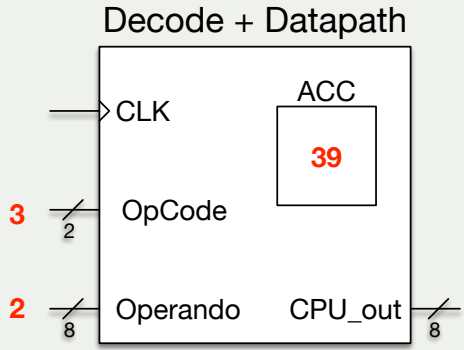
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

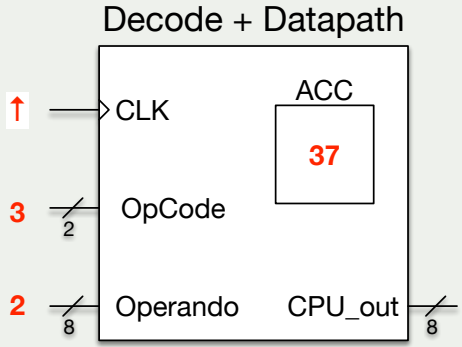
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

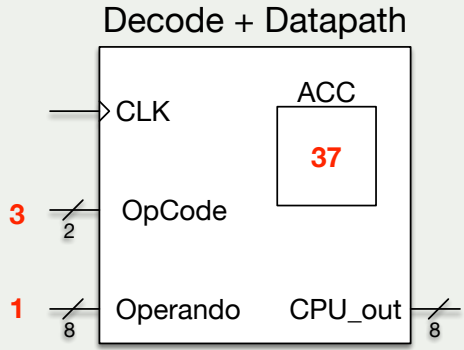
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

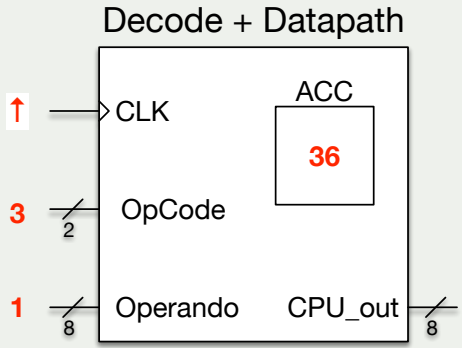
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

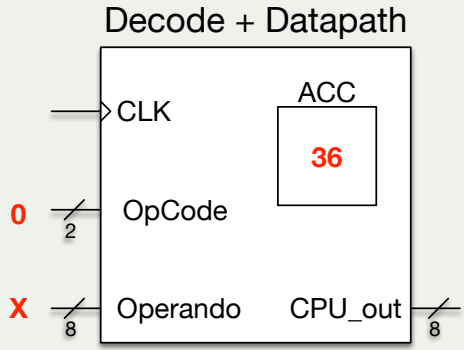
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

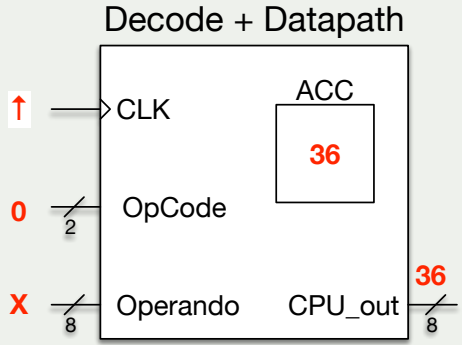
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Calculando com o Rudi

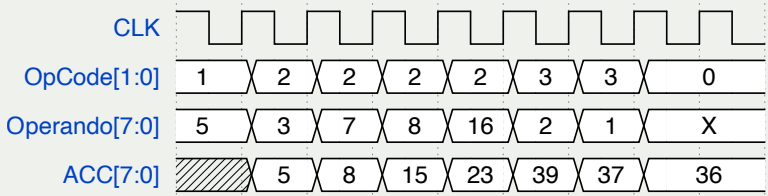
- H + I + J + K + L − N − O
- 5 + 3 + 7 + 8 + 16 − 2 − 1

Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



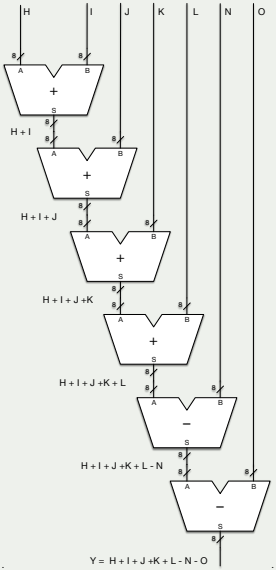
Calculando com o Rudi – Forma de onda

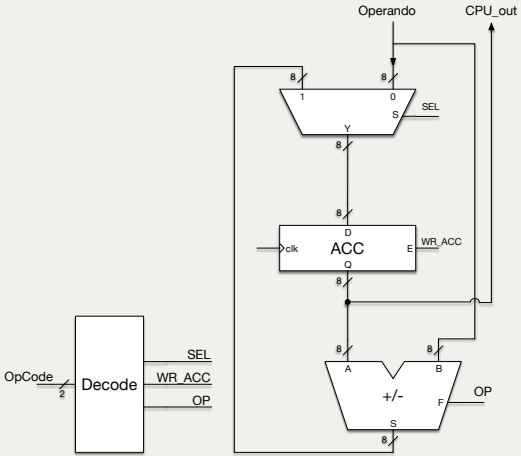
Operação	OpCode	Operando
$ACC \leftarrow 5$	01	00000101
$ACC \leftarrow ACC + 3$	10	00000011
$ACC \leftarrow ACC + 7$	10	00000111
$ACC \leftarrow ACC + 8$	10	00001000
$ACC \leftarrow ACC + 16$	10	00010000
$ACC \leftarrow ACC - 2$	11	00000010
$ACC \leftarrow ACC - 1$	11	00000001
$CPU\_out \leftarrow ACC$	00	XXXXXXXX



Comparação

Implementação Paralela vs. Rudi





Implementação Paralela vs. Rudi

- Implementação paralela:
  - 4 somadores e 2 subtratores
  - Entrada dos operandos em paralelo (pode ser mais rápida)
  - Fixa, apenas para a expressão projetada
- Rudi:
  - 1 somador, 2 multiplexadores, 1 decodificado e 1 registrador
  - Entrada dos operandos sequenciais
  - Flexível, pode implementar qualquer expressão com soma e subtração

FIM!