1. Publicação nº	2. Versão	3. Data	5. Distribuição
INPE-3458-RTR/069		*Março, 1985*	☐ Interna ☐ Externa ☒ Restrita
4. Origem: *DCA/DEA*	Programa: *SISMAG/PSDA*		

6. Palavras chaves - selecionadas pelo(s) autor(es)

UNIDADE ARITMÉTICA MICROPROGRAMADA
MICROPROGRAMAÇÃO

7. C.D.U.:

8. Título *INPE-3458-RTR/069*

ASTROM - UNIDADE ARITMÉTICA MICROPROGRAMADA

10. Páginas: *42*

11. Última página: *33*

12. Revisada por

León Lonneux

9. Autoria *Marcos Antonio Cardoso Cruz*

13. Autorizada por

Nelson de Jesus Parada
Diretor Geral

Assinatura responsável

14. Resumo/Notas

Descrevem-se as características gerais e o conjunto de instruções da unidade aritmética microprogramada ASTROM.

15. Observações

ABSTRACT

The general characteristics and instructions set of the ASTROM microprogrammed arithmetic unit are presented.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1

INTRODUÇÃO

As operações aritméticas são necessárias em sistemas de computação para aplicações em cálculo numérico, onde os números são comumente representados na notação inteira e na notação fracionária. Desta forma, torna-se necessária a implementação das operações básicas, tais como adição, subtração, multiplicação e divisão, quer seja através de programação ("software"), quer seja por meio de circuito ("hardware"). A diferença básica entre estas duas abordagens é o tempo de processamento envolvido.

Para aumentar a velocidade e a capacidade de cálculo numérico do computador ASTROP foi desenvolvida a unidade aritmética microprogramada ASTROM. Esta unidade executa 32 instruções de aritmética e lógica, operando com números inteiros (ponto fixo) de 16 bits e números fracionários (ponto flutuante) de 32 bits.

A sua estrutura de controle é do tipo microprogramada, baseada nos sequenciadores Am2909 e Am2911 (Advanced Micro Devices, 1977). A memória de controle, onde residem os microcódigos das rotinas que implementam as instruções do ASTROM, é do tipo estritamente horizontal e utiliza memórias não-voláteis (PROMs).

A parte de processamento e lógica utiliza componentes "bit-slice", de tecnologia Schottky e Low-Power Schottky (Advanced Micro Devices, 1976; Texas Instruments, 1978).

As especificações físicas/elétricas da unidade aritmética ASTROM são dadas na Tabela 1.1.

TABELA 1.1

ESPECIFICAÇÕES DA UNIDADE ARITMÉTICA

CARACTERÍSTICAS	ESPECIFICAÇÕES
- Tensão de alimentação	110V
- Consumo	60W
- Fusível	2A
- Ventilação	forçada
- Sinais de entrada:	
estado "0"	0,8V
estado "1"	2,0V
- sinais de saída:	
estado "0"	0,5V
estado "1"	2,5V
- Componentes de interfaceamento	TTL série 74
- Frequência de operação	5,5MHz
- Dimensões:	
altura	32,3mm
largura	47,4mm
profundidade	35,4mm
- Temperatura de operação	+ 15^{o}C a + 35^{o}C

CAPÍTULO 2

CONFIGURAÇÃO E INTERFACEAMENTO

A unidade aritmética microprogramada ASTROM é constituída por seis unidades funcionais, acondicionadas em uma caixa para uso em "rack" cuja montagem pode ser vista na Figura 2.1. As unidades funcionais são:

1) Placa P1: Unidade de Multiplicação e Interface (UML),

2) Placa P2: Unidade de Lógica e Processamento/Expoente (ULP/E),

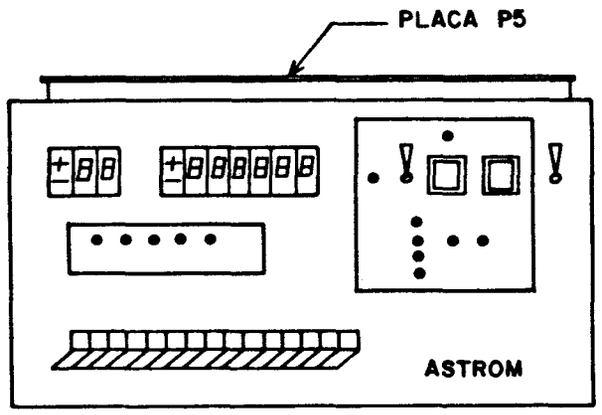
3) Placa P3: Unidade de Lógica e Processamento/Mantissa (ULP/M),

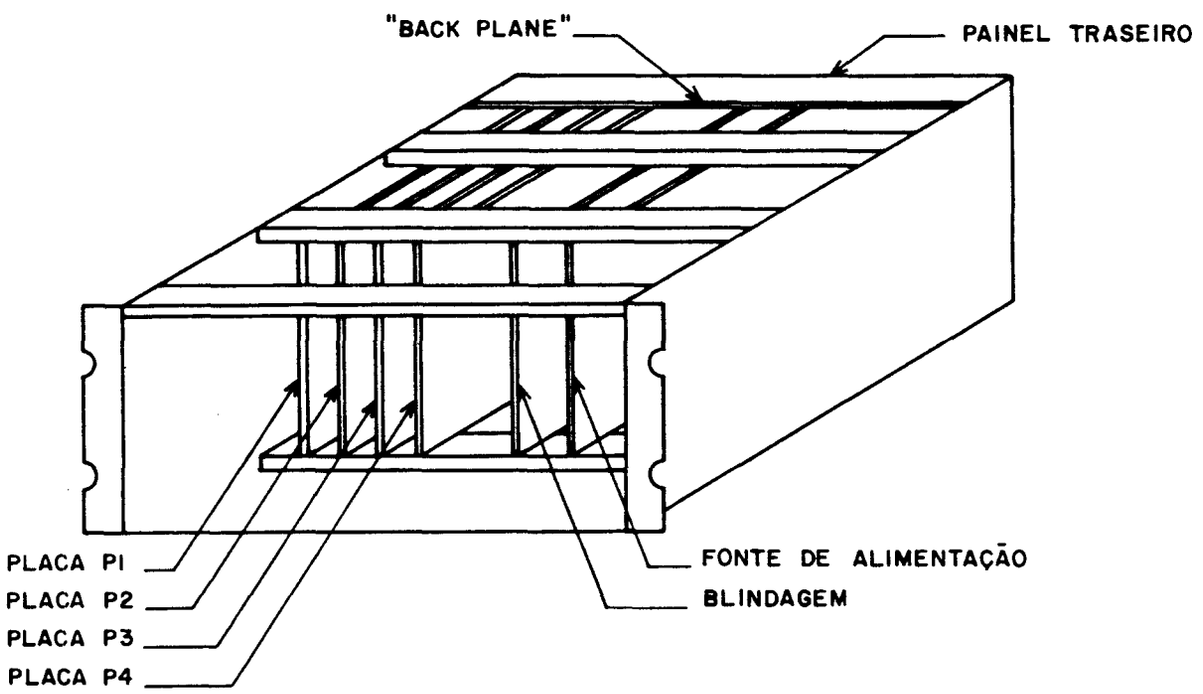
4) Placa P4: Unidade de Controle (UCN),

5) Placa P5: Painel do ASTROM,

6) Fonte de Alimentação.

A interconexão da unidade ASTROM com o computador ao qual ela estiver acoplada é feita através do conector K1, do tipo RS 50 pinos, montado no painel traseiro da caixa do ASTROM. A Tabela 2.1 contém a relação dos sinais neste conector.





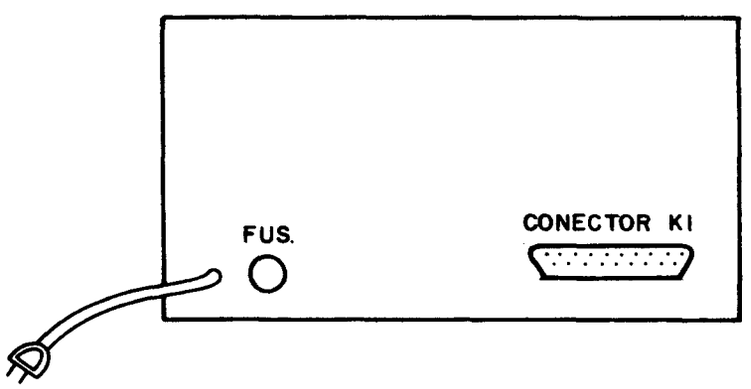


Fig. 2.1 - Vistas da caixa do ASTROM.

TABELA 2.1

SINAIS NO CONECTOR K1

SINAIS NOS CONECTORES	INPE - DCA/PSDA - PROG. DE SIST. DIGITAIS E ANALÓGICOS	FL: 1 DE 3
PLACA: CONECTOR K1 (ASTROM/UCP)		CÓD:
EQUIP: UNIDADE ARITMÉTICA ASTROM	PROJ: SISMAG	APROV: __/__/__ RESP:

PINO	SINAL	E/S	ORIGEM / DESTINO	DESCRIÇÃO	OBS.
1	-	-	-		
2	UPFOK*	S	UML	"Flag" de modo remoto	
3	ERCI*	S	ULP/M	"Flag" de erro de conversão	
4	OVF*	S	"	"Flag" de "overflow"	
5	DIVZ*	S	"	"Flag" de divisão por zero	
6	Z*	S	"	"Flag" de resultado igual a zero	
7	IDLEC	S	UCN	Sinalização de livre/ocupada	
8	-	-	-		
9	ASTC	E	UCP	Inicialização via computador	
10	D(14)	E/S	UML	D(15 - 0) - barramento de dados (15 - 0)	
11	D(12)	E/S	"		
12	D(10)	E/S	"		
13	D(8)	E/S	"		
14	D(6)	E/S	"		
15	D(4)	E/S	"		
16	D(2)	E/S	"		
17	D(0)	E/S	"		
18	-	-	-		
19	TERCI	S	GND	Terra do sinal ERCI	
20	UNDF*	S	ULP/M	"Flag" de "underflow"	
21	TDIVZ	S	GND	Terra do sinal DIVZ	
22	SIGN	S	ULP/M	"Flag" de sinal do resultado	
23	TIDLE	S	GND	Terra do sinal IDLE	
24	LERC	E	UML	Ler dados do ASTROM	

(continua)

Tabela 2.1 - Continuação

SINAIS NOS CONECTORES	INPE - DCA/PSDA - PROG. DE SIST. DIGITAIS E ANALÓGICOS	FL: 2 DE 3
PLACA: CONECTOR K1 (ASTROM/UCP)		CÓD:
EQUIP: UNIDADE ARITMÉTICA ASTROM	PROJ: SISMAG	APROV: __/__/__ RESP:

PINO	SINAL	E/S	ORIGEM / DESTINO	DESCRIÇÃO	OBS.
25	TRSTC	S	UML	Terra do sinal RSTC	
26	ATNC	E	UML	Requer atenção	
27	TRA4	S	GND	Terra	
28	-	-	-		
29	TRA3	S	GND	Terra	
30	-	-	-		
31	TRA2	S	GND	Terra	
32	-	-	-		
33	TRA1	S	GND	Terra	
34	-	-	-		
35	TUPFOK	S	GND	Terra do sinal UPFOK	
36	TUNDF	S	GND	Terra do sinal UNDF	
37	TOVF	S	GND	Terra do sinal OVF	
38	TSIGN	S	GND	Terra do sinal SIGN	
39	TZ	S	GND	Terra do sinal Z	
40	TLERC	S	UML	Terra do sinal TLERC	
41	-	-	-		
42	TATNC	S	UML	Terra do sinal ATNC	
43	D(15)	E/S	UML	D(15 - 0) - barramento de dados (15 - 0)	
44	D(13)	E/S	"		
45	D(11)	E/S	"		
46	D(9)	E/S	"		
47	D(7)	E/S	"		
48	D(5)	E/S	"		

(continua)

Tabela 2.1 - Conclusão

SINAIS NOS CONECTORES	INPE - DCA/PSDA - PROG. DE SIST. DIGITAIS E ANALÓGICOS	FL: 3 DE 3
PLACA: CONECTOR K1 (ASTROM/UCP)		CÓD:
EQUIP: UNIDADE ARITMÉTICA ASTROM	PROJ: SISMAG	APROV: __/__/__ RESP:

PINO	SINAL	E/S	ORIGEM / DESTINO	DESCRIÇÃO	OBS.
49	D(3)	E/S	UML		
50	D(1)	E/S	"		

A comunicação entre a unidade aritmética e o computador compreende os seguintes sinais:

1) barramento de dados bidirecional D(15-ϕ) com 16 linhas;

2) sinais de controle IDLE, ATNC e LERC, cujo "handshake" é mostrado nas Figuras 2.2 e 2.3;

3) "flags" do ASTROM, constituídos pelos sinais Z, SIGN, UNDF, OVF, DVIZ e ERCI, cuja descrição é dada na Tabela 2.2.

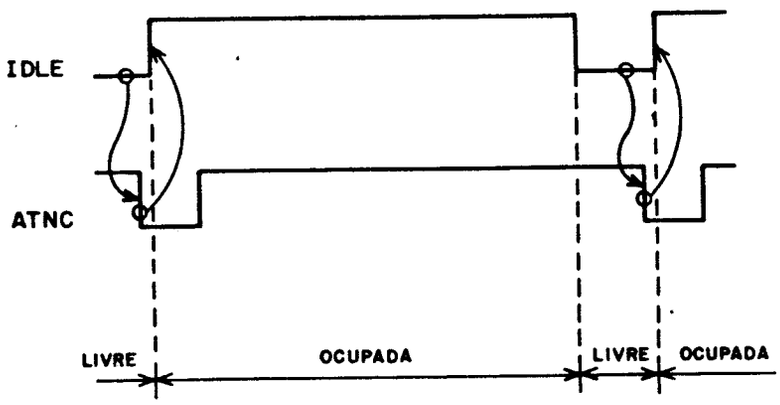


Fig. 2.2 - Diagrama de sincronização ASTROM/UCP.

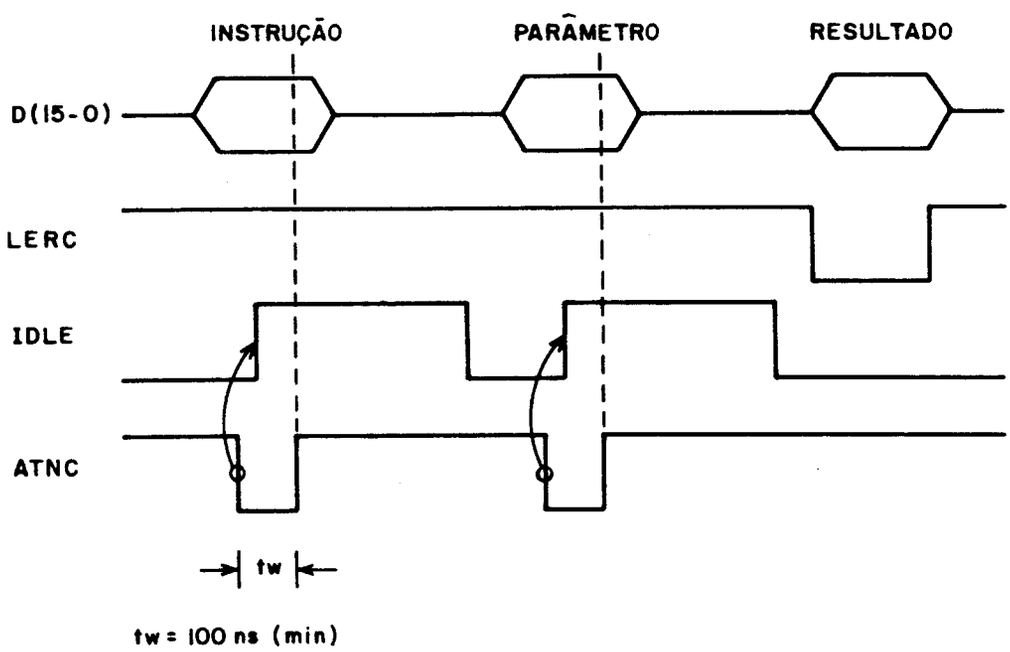


Fig. 2.3 - Transferência de dados entre ASTROM e UCP.

TABELA 2.2

LISTAGEM DOS "FLAGS" DO ASTROM

"FLAG"	ATIVO	DESCRIÇÃO
Z	L	Resultado igual a zero
SIGN	-	Sinal do resultado {L - se positivo; H - se negativo}
UNDF	L	"Underflow" do resultado
OVF	L	"Overflow" do resultado
DIVZ	L	Divisão por zero
ERCI	L	Erro de conversão de flutuante para inteiro

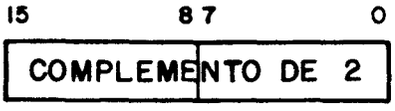
OBS.: L - nível baixo;
H - nível alto.

A seleção do modo de operação do ASTROM, localmente através do Painel ou remotamente através do computador, é feita por meio da chave UCP/Painel existente no painel frontal da caixa do ASTROM. Quando esta chave está na posição UCP, a operação é feita com o computador; quando está na posição Painel, é feita pelo usuário, através do Painel.

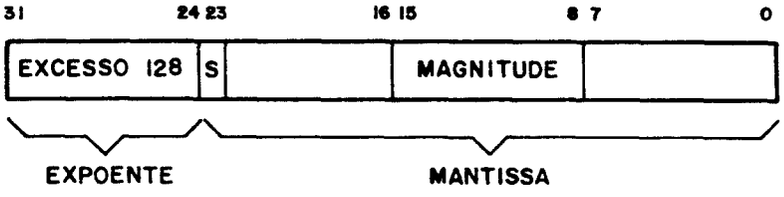
CAPÍTULO 3

PROGRAMAÇÃO DO ASTROM

A unidade aritmética ASTROM opera com números em ponto fixo (inteiro) e com números em ponto flutuante (fracionário), cuja representação é dada na Figura 3.1. Os números em ponto fixo utilizam 16 bits representados em complemento de 2. Os números em ponto flutuante utilizam 32 bits, dos quais 8 correspondem ao expoente e 24 correspondem à mantissa. O expoente é representado em excesso de 128, enquanto a mantissa utiliza a representação de sinal e magnitude.



(a) - número em ponto fixo



(b) - número em ponto flutuante

Fig. 3.1 - Representação dos números no ASTROM.

Os números em ponto fixo variam de - 32768 a +32767, conforme a Tabela 3.1.

TABELA 3.1

VALOR DOS NÚMEROS EM PONTO FIXO

VALOR HEXADECIMAL (COMPLEMENTO DE 2)	VALOR DECIMAL
7FFF	+32767
.	.
.	.
.	.
0002	+2
0001	+1
0000	0
FFFF	-1
FFFE	-2
.	.
.	.
.	.
8000	-32768

O expoente do número em ponto flutuante varia de -127 a 127 na base 2, cujo valor na base 10 varia de -38 a +38, aproximadamente. A mantissa é representada na forma normalizada, e o primeiro bit após o ponto da fração é igual a "1"; por isso ele é omitido na representação da mantissa. A Figura 3.2 ilustra a representação do número $+0,6825 \times 2^{+5}$.

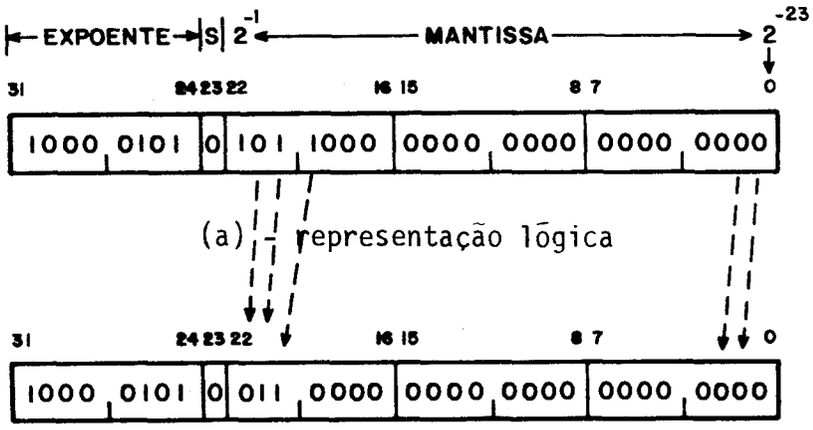


Fig. 3.2 - Representação em ponto flutuante.

Na representação em ponto flutuate os bits da mantissa têm peso 2^{-1}, 2^{-2}, 2^{-3}... etc. O bit mais significativo à esquerda tem peso 2^{-1} e o bit menos significativo à direita tem peso 2^{-23}. Desta forma, a mantissa normalizada tem um valor x tal que $1/2 \leq x < 1$. O sinal da mantissa será "0" se ela for positiva e "1" se for negativa.

A Tabela 3.2 mostra a faixa de valores do expoente. cuja representação é obtida somando 128 (80 em hexadecimal) ao valor em complemento de 2 do número desejado. Assim, -5 em complemento de 2 é FB_{16} e em excesso de 128 é EB_{16}. O valor 00 é reservado para representar o número 0×2^0.

TABELA 3.2

VALOR DO EXPOENTE EM PONTO FLUTUANTE

VALOR HEXADECIMAL (EXCESSO DE 128)	VALOR DECIMAL
FF	+127
.	.
.	.
.	.
81	+1
80	0
EF	-1
EE	-2
.	.
.	.
.	.
01	-127
00	$0x2^0$

A unidade aritmética ASTROM possui 8 registros de dados, RØ a R7, dos quais quatro são acessíveis ao usuário (RØ a R3) e os outros quatro são utilizados internamente nas rotinas que implementam as funções do ASTROM. Estes registros possuem 32 bits, dos quais os 16 mais significativos são zerados quando se opera com números em ponto fixo.

As instruções do ASTROM fazem parte do conjunto de instrução do computador ASTROP, num total de 32 instruções, cujas operações podem ser classificadas em:

1) operações aritméticas, que compreendem a adição, subtração, multiplicação e divisão tanto em ponto fixo como em ponto flutuante;

2) operações de lógica constituídas pelas instruções de negação, valor absoluto em ponto fixo e flutuante e comparação em ponto flutuante;

3) operações de conversão para mudança de representação de ponto fixo para flutuante e vice-versa;

4) operações de trasferência, que compreendem as instruções de transferência de dados entre: a unidade artmética e o computador, a unidade aritmética e o painel, e internamente à ASTROM.

As instruções do ASTROM podem ser de uma, duas ou três palavras de 16 bits cadas uma, conforme o tipo da instrução. Basicamente têm-se 5 tipos de instruções, ou seja:

1) instrução sem transferência de dados,

2) instrução com operando 16 bits,

3) instrução com operando de 32 bits,

4) instrução com resultado de 16 bits,

5) instrução com resultado de 32 bits.

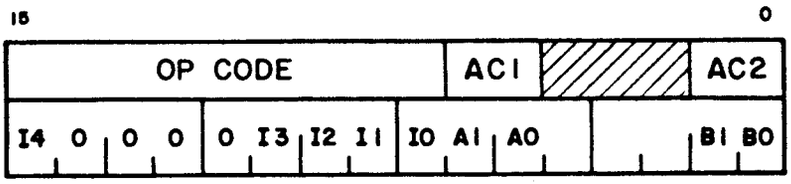
Os formatos das instruções, mostrados na Figura 3.3, são constituídos de:

- um campo de código de operação (OPCODE);

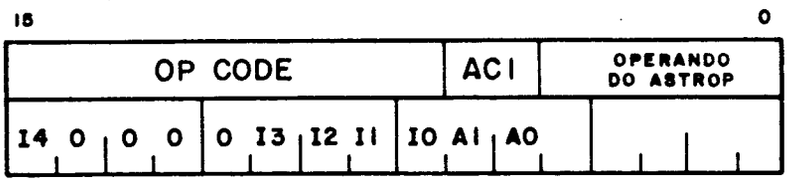
- um campo de endereço do registro destino (AC1);

- um campo de endereço - do registro fonte (AC2) se for uma instrução sem transferência de dados;

  - do operando fonte no ASTROP se for uma instrução com operando ou com resultado.



(a) - instrução sem transferência de operando



(b) - instrução com operando ou resultado

Fig. 3.3 - Formatos das instruções do ASTROM.

Nas instruções com operando ou resultado de 16 bits, o dado é tipicamente de ponto fixo, da mesma forma que nas instruções com operando ou resultado de 32 bits, o dado é de ponto flutuante. No caso de um dado de 32 bits, a primeira palavra de dado contém os 16 bits menos significativos da mantissa e a segunda palavra de dado contém os 8 bits mais significativos da mantissa e os 8 bits do expoente.

A seguir serão descritas as instruções da unidade aritmética ASTROM, de acordo com as suas classes de operação, e ressaltados os "flags" afetados pela execução da instrução.

3.1 - INSTRUÇÕES ARITMÉTICAS

As instruções aritméticas do ASTROM são:

1) SMIM
2) SMIP
} soma inteiro

3) SUIM
4) SUIP
} subtrai inteiro

5) MLIM } multiplica inteiro
6) MLIP }

7) DVIM } divide inteiro
8) DVIP }

9) SMPFM } soma flutuante
10) SMPFP }

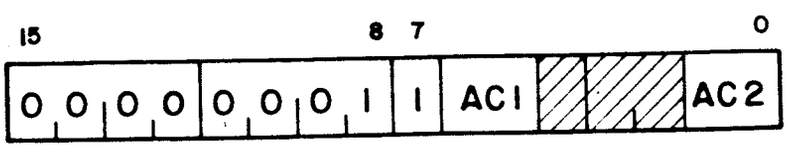
11) SUPFM } subtrai flutuante
12) SUPFP }

13) MLPFM } multiplica flutuante
14) MLPFP }

15) DVPFM } divide flutuante
16) DVPFP }

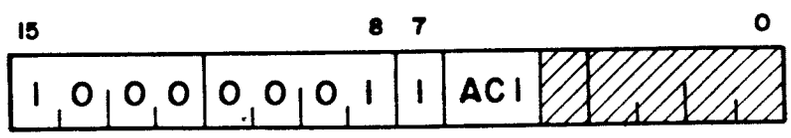
a) Soma Inteiro

- SMIM



operação: (AC1) ← (AC1) + (AC2)
tempo de execução: 1.980ns

- SMIP



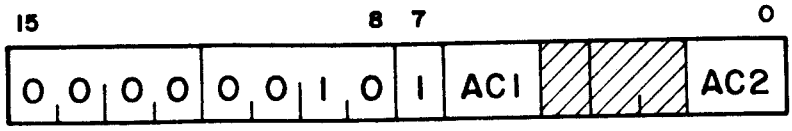
operação: (AC1) ← (AC1) + dado de 16 bits
tempo de execução: 2.520ns

- "Flags" modificados:

SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

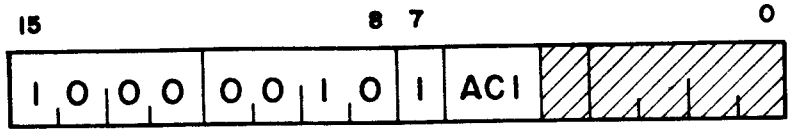
b) Subtrai Inteiro

- SUIM



operação: (AC1) ← (AC1) - (AC2)
tempo de execução: 1.980ns

- SUIP



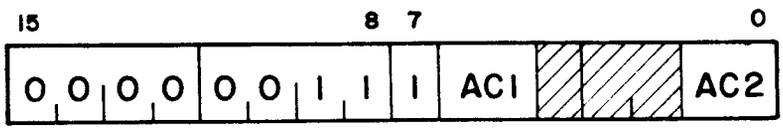
operação: (AC1) ← (AC1) - dado de 16 bits
tempo de execução. 2.520ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

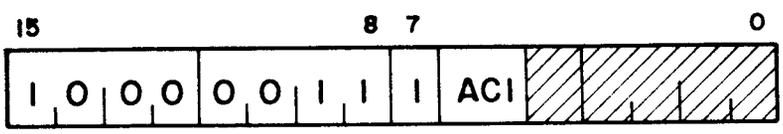
c) Multiplica Inteiro

- MLIM



operação: (AC1) ← (AC1) x (AC2)
tempo de execução: 5.400ns

- MLIP



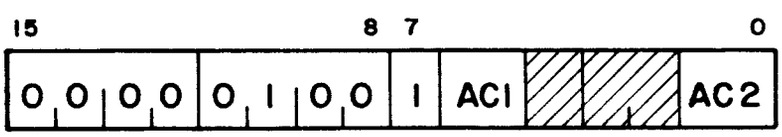
operação: (AC1) ← (AC1) x dado de 16 bits
tempo de execução: 5.940ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

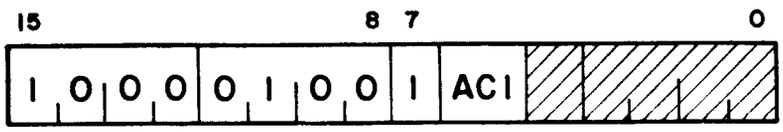
d) Divide Inteiro

- DVIM



operação: (AC1) ← (AC1) ÷ (AC2)
tempo de execução: 19.080ns

- DVIP



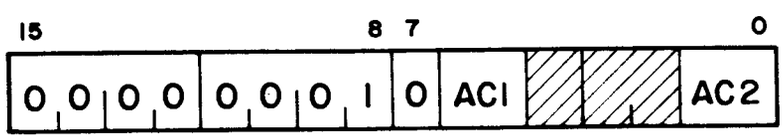
operação: (AC1) ← (AC1) ÷ dado de 16 bits
tempo de execução: 19.620ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0
DIVZ ← 0/1: normal/divisão por zero
ERCI ← 0

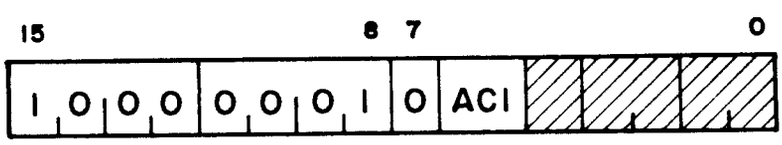
e) Soma Flutuante

- SMPFM



operação: (AC1) ← (AC1) + (AC2)
tempo de execução: 5.220ns

- SMPFP



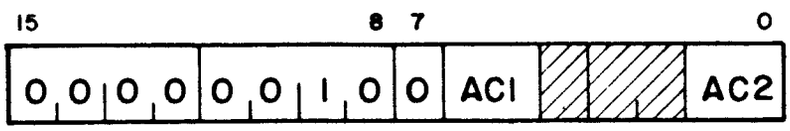
operação: (AC1) ← (AC1) + dado de 32 bits
tempo de execução: 6.120ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0/1: não-"underflow"/"underflow"
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

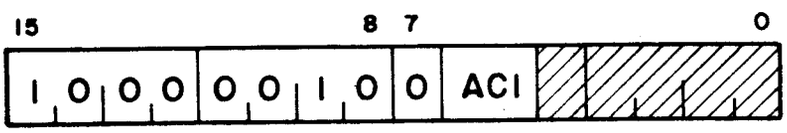
f) Subtrai Flutuante

- SUPFM



operação: (AC1) ← (AC1) - (AC2)
tempo de execução: 7.020ns

- SUPFP



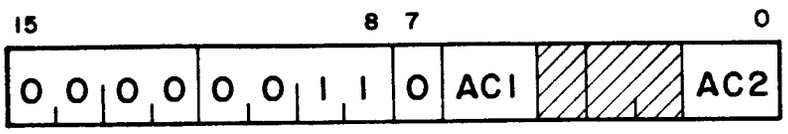
operação: (AC1) ← (AC1) - dado de 32 bits
tempo de execução: 7.920ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0/1: não-"underflow"/"underflow"
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

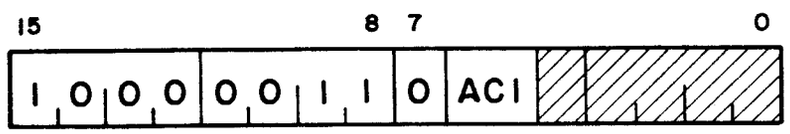
g) Multiplica Flutuante

- MLPFM



operação: (AC1) ← (AC1) x (AC2)
tempo de execução: 3.780ns

- MLPFP



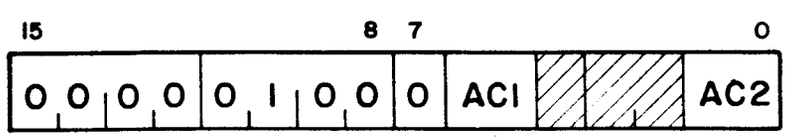
operação: (AC1) ← (AC1) x dado de 32 bits
tempo de execução: 4.680ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0/1: não-"underflow"/"underflow"
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

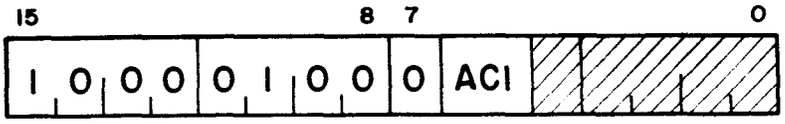
h) Divide Flutuante

- DVPFM



operação: (AC1) ← (AC1) ÷ (AC2)
tempo de execução: 12.600ns

- DVPFP



operação: (AC1) ← (AC1) ÷ dado de 32 bits
tempo de execução: 13.500ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0/1: n~ao-"underflow"/"underflow"
OVF ← 0
DIVZ ← 0/1: normal/divisão por zero
ERCI ← 0

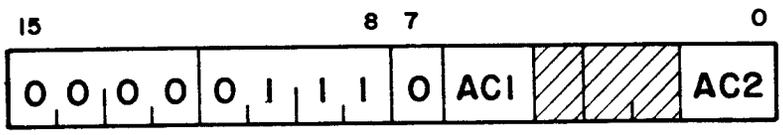
3.2 - INSTRUÇÕES DE LÓGICA

As instruções de lógica do ASTROM são:

1) NGIN - negação em ponto fixo
2) NGPFM } negação em ponto flutuante
3) NGPFP }
4) ABSI - valor absoluto em ponto fixo
5) ABPF - valor absoluto em ponto flutuante
6) CMPFM } comparação em ponto flutuante
7) CMPFP }

a) Negação em Ponto Fixo

- NGIN



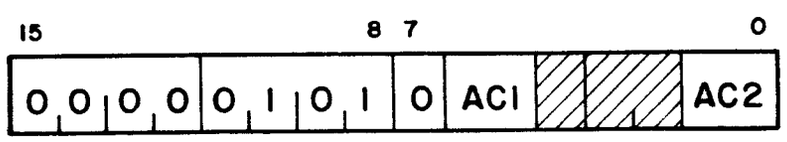
operação: (AC1) ← - (AC2)
tempo de execução: 2.160ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

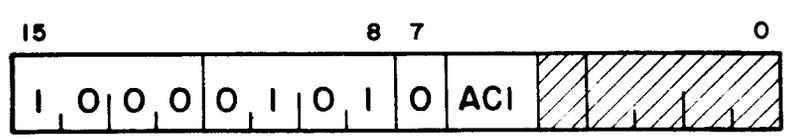
b) Negação em Ponto Flutuante

- NGPFM



operação: (AC1) ← - (AC2)
tempo de execução: 1.800ns

- NGPFP



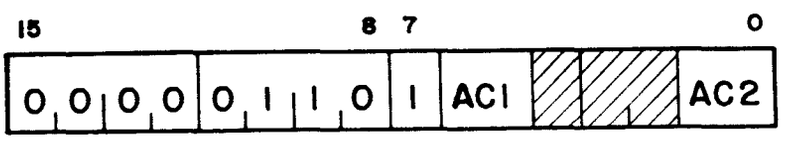
operação: (AC1) ← - (dado de 32 bits)
tempo de execução: 2.700ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

c) Valor Absoluto em Ponto Fixo

- ABSI



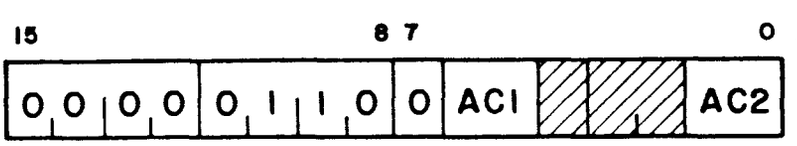
operação: (AC1) ← /(AC2)/
tempo de execução: 1.620ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0: positivo
UNDF ← 0
OVF ← 0/1: não-"overflow"/"overflow"
DIVZ ← 0
ERCI ← 0

d) Valor Absoluto em Ponto Flutuante

- ABPF



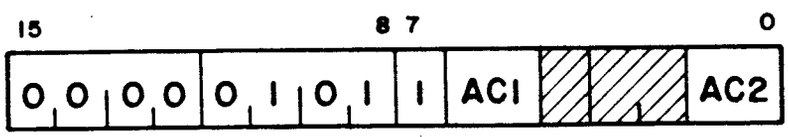
operação: (AC1) ← /(AC2)/
tempo de execução: 1.260ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 1
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

e) Comparação em Ponto Flutuante

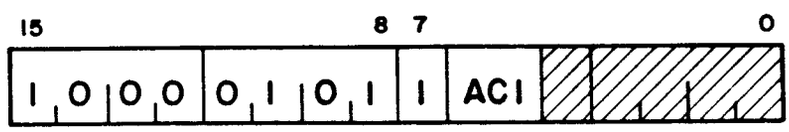
- CMPFM



operação: (AC1) - (AC2)
tempo de execução: 2.880ns

OBS.: Esta instrução não altera o conteúdo do registro AC1.

- CMPFP



operação: (AC1) - (dado de 32 bits)
tempo de execução: 3.780ns

OBS.: Esta instrução não altera o conteúdo do registro AC1.

- "Flags" modificados:

Z ← 1 e SIGN ← 0: se (AC1) = (AC2), no caso de CMPFM;
se (AC1) = dado, no caso de CMPFP.

Z ← 0 e SIGN ← 0: se (AC1) > (AC2), no caso de CMPFM;
se (AC1) > dado, no caso de CMPFP.
Z ← 0 e SIGN ← 1: se (AC1) < (AC2), no caso de CMPFM;
se (AC1) < dado, no caso de CMPFP.
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

3.3 - INSTRUÇÕES DE CONVERSÃO

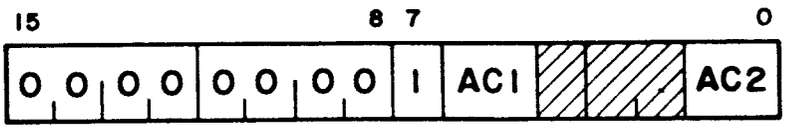
As instruções de conversão do ASTROM são:

1) IPFM
2) IPFP
} convertem ponto fixo em ponto flutuante;

3) PFIM
4) PFIP
} convertem ponto flutuante em ponto fixo.

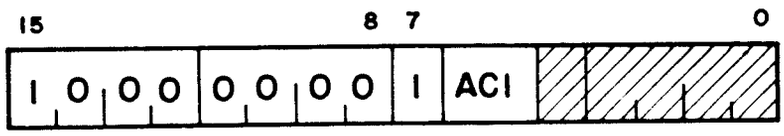
a) Convertem Ponto Fixo em Ponto Flutuante:

- IPFM



operação: (AC1) [FLUTUANTE] ← (AC2) [FIXO]
tempo de execução: 4.140ns

- IPFP



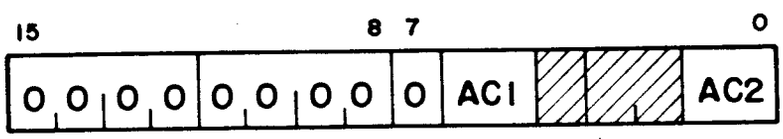
operação: (AC1) [FLUTUANTE] ← dado de 16 bits [FIXO]
tempo de execução: 4.680ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

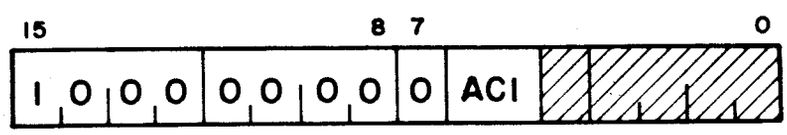
b) Convertem Ponto Flutuante em Ponto Fixo:

- PFIM



operação: (AC1) [FIXO] ← (AC2) [FLUTUANTE]
tempo de execução: 4.320ns

- PFIP



operação: (AC1) [FIXO] ← dado de 32 bits [FLUTUANTE]
tempo de execução: 5.220ns

- "Flags" modificados:

Z ← 0/1: não-zero/zero
SIGN ← 0/1: positivo/negativo
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0/1: conversão possível/conversão impossível

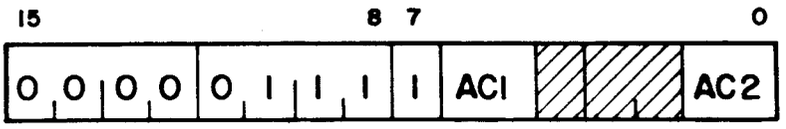
3.4 - INSTRUÇÕES DE TRANSFERÊNCIA

As instruções de transferência do ASTROM são:

1) CPRGI - copia registro interno
2) CRI - carrega inteiro
3) CRPF - carrega ponto flutuante
4) AZI - armazena inteiro
5) AZPF - armazena ponto flutuante

a) Copia Registro Interno

- CPRGI



operação: (AC1) ← (AC2)
tempo de execução: 1.080ns

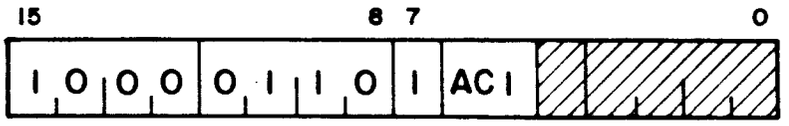
- "Flags" modificados:

Z ← 0
SIGN ← 0
UNDF ← 0

OVF ← 0
DIVZ ← 0
ERCI ← 0

b) Carrega Inteiro

- CRI



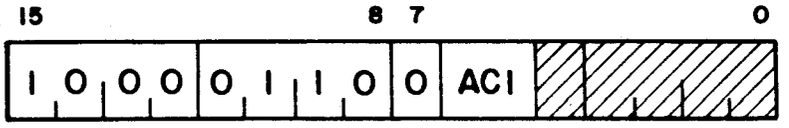
Operação: (AC1) ← dado de 16 bits
tempo de execução: 1.800ns

- "Flags" modificados:

Z ← 0
SIGN ← 0
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

c) Carrega Ponto Flutuante

- CRPF



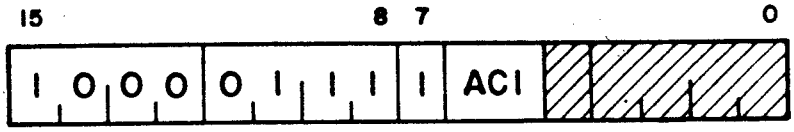
operação: (AC1) ← dado de 32 bits
tempo de execução: 2.160ns

- "Flags" modificados:

Z ← 0
SIGN ← 0
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

d) Armazena Inteiro

- AZI



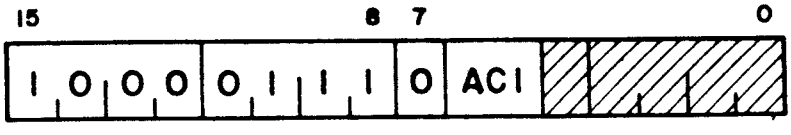
operação: dado de 16 bits [UCP/Painel] ← (AC1)
tempo de execução: 1.260ns

- "Flags" modificados:

Z ← 0
SIGN ← 0
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

e) Armazena Ponto Flutuante

- AZPF



operação: dado de 32 bits [UCP/Painel] ← (AC1)
tempo de execução: 2.160ns

- "Flags" modificados:

Z ← 0
SIGN ← 0
UNDF ← 0
OVF ← 0
DIVZ ← 0
ERCI ← 0

3.5 - OBSERVAÇÕES GERAIS

Algumas observações referentes à execução das instruções do ASTROM que precisam ser consideradas para efeito de programação são:

1) Todos os operandos em ponto fixo após a execução da instrução são armazenados nos registros de dados com os 8 bits do expoente e os 8 bits mais significativos da mantissa que contêm "0";

2) Toda instrução cuja execução resulte em uma condição de erro, ou seja, "overflow", "underflow", divisão por zero ou erro de conversão para inteiro, armazenará um resultado com todos os bits iguais a "0"; neste caso o "flag" Z também será sinalizado.

3) No cálculo do tempo de execução das instruções, foram considerados os tempos relativos ao ASTROM, não sendo computados os tempos de busca de operandos da UCP/Painel que por ventura sejam necessários para a execução da instrução.

REFERÊNCIAS BIBLIOGRÁFICAS

ADVANCED MICRO DEVICES (AMD). *The Am 2900 family data book.* Sunnyvale, CA, 1977.

——— *Schottly and low-power schotthy data book.* Synnyvale, CA, 1976.

TEXAS INSTRUMENTS INC. *The TTL book for design engineers.* Dallas, TX, 1978.