2　記憶装置を理解しよう

2-1 主記憶装置に値を保存する
2-2 主記憶装置の値を読み込む
2-3 主記憶装置の値を別の場所にコピーする
2-4 LD命令とLAD命令
2-5 キーボードから値を入力する

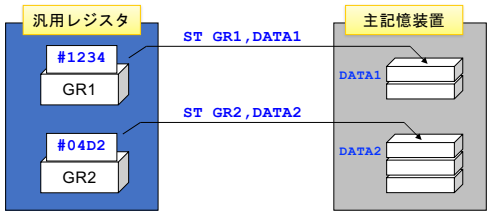
2-1 主記憶装置に値を保存する(0)

```
 1：PRG0201 START
 2：;         RPUSH
 3：          LAD GR1,#1234
 4：          LAD GR2,1234
 5：          ST GR1,DATA1     ;GR1の値をDATA1に格納する
 6：          ST GR2,DATA2     ;GR2の値をDATA2に格納する
 7：;         RPOP
 8：          RET
 9：DATA1     DS 2             ;2語分の領域（場所）を確保
10：DATA2     DS 3             ;3語分の領域（場所）を確保
11：          END
```

	GR0	GR1	GR2	GR3	DATA1		DATA2		
(a) 3行目実行前	0000	0000	0000	0000	0000	0000	0000	0000	0000
(b) 3行目実行後	0000	1234	0000	0000	0000	0000	0000	0000	0000
(c) 4行目実行後	0000	1234	04D2	0000	0000	0000	0000	0000	0000
(d) 5行目実行後	0000	1234	04D2	0000	1234	0000	0000	0000	0000
(e) 6行目実行後	0000	1234	0000	0000	1234	0000	04D2	0000	0000

2-1 主記憶装置に値を保存する(1)

■ST命令の動作様子



2-1 主記憶装置に値を保存する(2)

- ST命令（ストア命令）は，
  - 汎用レジスタに設定されている値を主記憶装置という別の記憶装置へ格納（コピー）する命令である．
  - オペランドでは，汎用レジスタは8個のうちのどれを使うのか，格納（コピー）先は主記憶装置のどの場所なのかを示している．

命令	書き方		命令の説明
	命令コード	オペランド	
ストア Store	ST	r,adr	adr ← (r) rはGR0～GR7

2-1 主記憶装置に値を保存する(3)

- 主記憶装置
  - 実行するプログラム自身や値を記憶しておいたりする場所が主記憶装置である
  - なぜ，必要か？
    - CPUの内部に記憶装置として，汎用レジスタがありが8個しかない
    - 多くの値や文字列などプログラムの処理に不十分
  - 汎用レジスタと主記憶の一番の大きな違いは，
    - 値を記憶できる量（記憶容量）である．
    - COMETⅡでは主記憶の記憶容量は65,536語（64キロ語）である．
    - COMETⅡでは16ビット単位でデータを取扱うので，1語は16ビットである．

39

39

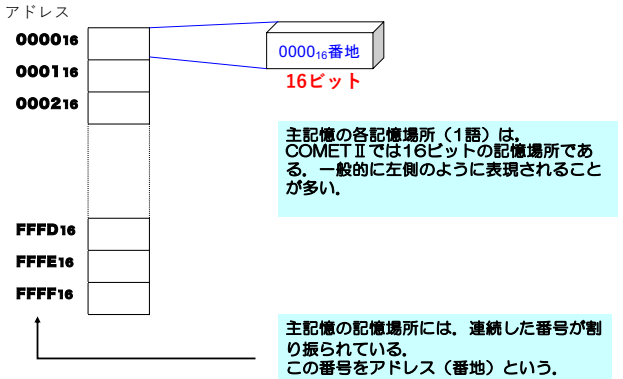
2-1 主記憶装置に値を保存する(4)

- 主記憶装置の特徴
  - 大きな量の主記憶なので，特定の場所を指定するために，主記憶の各箱にアドレスと呼ばれる数値が割り振られている．
    - COMETⅡではアドレスは0～65,535（0000_{16}～$FFFF_{16}$）番地である．
  - プログラムから主記憶にアクセスするときには，このアドレスを指定してアクセスすることになる．
  - アセンブラ言語では，アドレスを管理する方法としてラベルが利用できる．
  - ラベルは
    - アセンブルして機械語に変換すると，自動的に具体的なアドレス（具体的な数値）に変換される．

40

40

2-1 主記憶装置に値を保存する(5)



41

41

2-1 主記憶装置に値を保存する(6)

- DS命令

【9,10行目】

```
DATA1   DS   2
DATA2   DS   3
```

  - 9行目の命令は，主記憶上に，DATA1というラベルで2語分の領域を確保する命令である．
  - 10行目の命令は，主記憶上に，DATA2というラベルで3語分の領域を確保する命令である．
  - DS命令は，オペランドで指定した語数だけ，主記憶上に領域を確保する命令である．

命令の種類	書き方	機能
アセンブラ命令	[ラベル] DS 語数	指定した語数の領域を確保する

■注意点

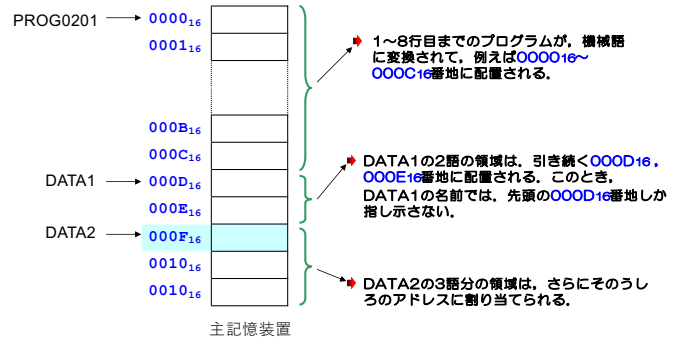
```
        RET
DATA1   DS   2
DATA2   DS   3
        END
```

DS, DCは，RETとENDの間

42

42

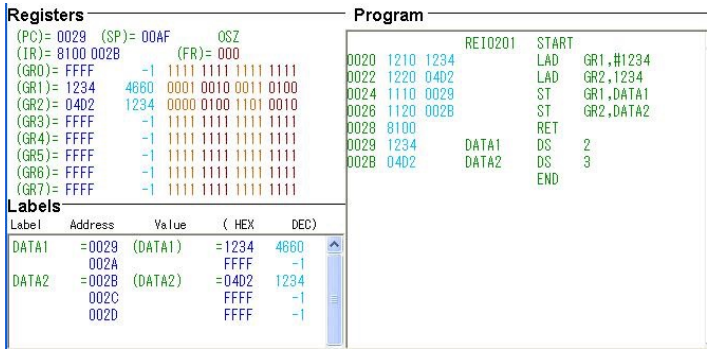
2-1 主記憶装置に値を保存する(7)



43

2-1 主記憶装置に値を保存する(8)

■WCASL2による実行結果



44

2-2 主記憶装置の値を読み込む(0)

```
 1：PRGO202 START
 2：         RPUSH
 3：         LAD GRO,65535
 4：         ST GRO,DATA1
 5：         LD GR1,DATA1     ；主記憶のDATA1の値を読込む
 6：         LD GR2,DATA2
 7：         LD GR3,DATA3
 8：         LD GR4,DATA4
 9：         ST GR4,DATA2     ；定数定義した領域に値を格納する
10：         RPOP
11:        RET
12：DATA1   DS  1
13：DATA2   DC  100
14：DATA3   DC  #0100,#0101
15：DATA4   DC  '0123456789'
16：        END
```


45

2-2 主記憶装置の値を読み込む(1)

	GR0	GR1	GR2	GR3	GR4	DATA1	DATA2	DATA3		DATA4		
(a)3行目実行前	0000	0000	0000	0000	0000	0000	0064	0100	0101	0030	0031	…
(b)3行目実行後	FFFF	0000	0000	0000	0000	0000	0064	0100	0101	0030	0031	…
(c)4行目実行後	FFFF	0000	0000	0000	0000	FFFF	0064	0100	0101	0030	0031	…
(d)5行目実行後	FFFF	FFFF	0000	0000	0000	FFFF	0064	0100	0101	0030	0031	…
(e)6行目実行後	FFFF	FFFF	0064	0000	0000	FFFF	0064	0100	0101	0030	0031	…
(f)7行目実行後	FFFF	FFFF	0064	0100	0000	FFFF	0064	0100	0101	0030	0031	…
(g)8行目実行後	FFFF	FFFF	0064	0100	0030	FFFF	0064	0100	0101	0030	0031	…
(h)9行目実行後	FFFF	FFFF	0064	0100	0030	FFFF	0030	0100	0101	0030	0031	…

46

2-2 主記憶装置の値を読み込む(2)

- 主記憶装置へ値を設定

【3,4,12行目】　`LAD　　GR0,65535`

- 3，4行目を実行すると，レジスタGR0と主記憶のDATA1には，ともに$FFFF_{16}$（10進数の65,535）が設定される［実行のようす（b），（c）］．

- LD命令

【5行目】　`LD　　GR1,DATA1`

- LD命令は，主記憶のDATA1の場所に格納されている値を汎用レジスタGR1に設定（コピー）する．
- DATA1には，4行目のST命令で$FFFF_{16}$が格納されていたので，GR1にも$FFFF_{16}$が設定される［実行のようす（d）］．

47

47

2-2 主記憶装置の値を読み込む(3)

- LD命令（ロード命令）とは，
  - 主記憶のadrの場所に格納されている値を汎用レジスタrに設定（コピー）する命令である．

命令	書き方		命令の説明
	命令コード	オペランド	
ロード LoaD	LD	r,adr	r ← (adr) rはGR0～GR7

48

48

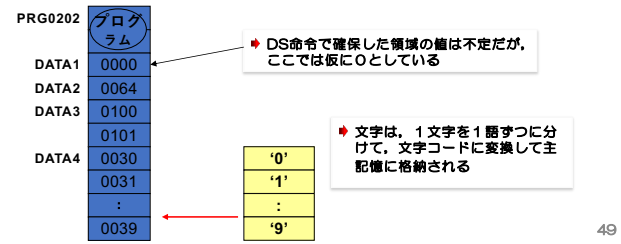
2-2 主記憶装置の値を読み込む(4)

- DC命令

【13行目】　`DATA2　DC　　100`

- 主記憶のDATA2という場所に値100（0064₁₆）がプログラム実行開始時に設定される［実行のようす（a）］．
- DC命令は，オペランドに書かれた値そのものを，プログラムの実行開始時に1回だけ主記憶に設定する命令である．

命令の種類	書き方	機能
アセンブラ命令	［ラベル］　DC　定数［,定数］…	定数を定義



49

49

2-2 主記憶装置の値を読み込む(5)

- DC命令

【14行目】　`DATA3　DC　　#0100,#0101`

- 複数の値を設定しておきたいとき，その数値を“，”で書き並べることができる．

【15行目】　`DATA4　DC　　‘0123456789’`

- シングルコーテーション（‘　’）で囲った文字は，文字列0123456789として定義される．
- その文字が文字コードに変換され，1文字が1語（主記憶の1つの領域の大きさ）になるようにして主記憶に格納される．

［文字］	［文字コード］	［1語で表現された文字コード］
‘0’ →	30_{16} →	0030_{16}
‘1’ →	31_{16} →	0031_{16}

- 注意
  - DC命令は，定数定義命令と呼ばれいるが，値を書き換えることが出来てしまうので，注意が必要である［9行目］．

50

50

問題[2-2]

まず、ST命令を使用し，主記憶のDAT領域（ラベル名DAT）に値58_{16}を格納し，次に、その値をGR1，GR3に読み出すプログラムを作成しなさい．

51

51

2-3 主記憶装置の値を別の場所にコピーする(1)

✓ プログラムPRG0203は，主記憶のORG1～ORG4に格納されている値を，DST1～DST4にコピーするプログラムである．

```
 1 PRG0203  START
 2          RPUSH
 3          LD      GR1,ORG1
 4          ST      GR1,DST1     ；ORG1の値をDST1にコピー
 5          LD      R1,ORG2
 6          ST      GR1,DST2     ；ORG2の値をDST2にコピー
 7          LD      GR1,ORG3
 8          LD      GR2,ORG4
 9          ST      GR1,DST3     ；ORG3の値をDST3にコピー
10          ST      GR2,DST4     ；ORG4の値をDST4にコピー
11          LD      GR3,GR2
12          RPOP
13          RET
14 ORG1     DC      '1'
15 ORG2     DC      '2'
16 ORG3     DC      '3'
17 ORG4     DC      '4'
18 DST1     DS      1
19 DST2     DS      1
20 DST3     DS      1
21 DST4     DS      1
22          END
```

52

52

2-3 主記憶装置の値を別の場所にコピーする(1)

	汎用レジスタ			主記憶装置			
	GR1	GR2	GR3	DST1	DST2	DST3	DST4
(a) 3行目実行前	0000	0000	0000	0000	0000	0000	0000
(b) 3行目実行後	0031	0000	0000	0000	0000	0000	0000
(c) 4行目実行後	0031	0000	0000	0031	0000	0000	0000
(d) 5行目実行後	0032	0000	0000	0031	0000	0000	0000
(e) 6行目実行後	0032	0000	0000	0031	0032	0000	0000
(f) 7行目実行後	0033	0000	0000	0031	0032	0000	0000
(g) 8行目実行後	0033	0034	0000	0031	0032	0000	0000
(h) 9行目実行後	0033	0034	0000	0031	0032	0033	0000
(i) 10行目実行後	0033	0034	0000	0031	0032	0033	0034
(j) 11行目実行後	0033	0034	0034	0031	0032	0033	0034

53

53

2-3 主記憶装置の値を別の場所にコピーする(2)

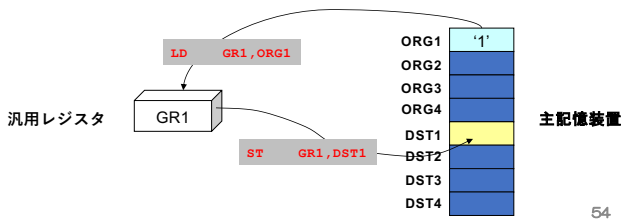
- **LD命令とST命令でコピーする**

【3,14行目】 `LD GR1,ORG1`

- 3行目は，LD命令でORG1に格納されているデータをGR1に設定する．
- 14行目は，DC命令で，ORG1に'1'（文字コード0031_{16}）と定義しているので，GR1には値0031_{16}が格納される［実行のようす（b）］．

【4,18行目】 `ST GR1,DST1`

- 4行目は，ST命令でGR1に設定している値0031_{16}を主記憶のDST1に格納する［実行のようす（c）］．
- 18行目は，DS命令でDST1に1語分の領域として定義している．
- 3行目のLD命令と，4行目のST命令で，ORG1に格納されていた値0031_{16}をDST1にコピーすることができた．



54

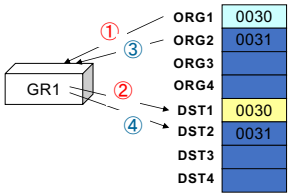
54

2-3 主記憶装置の値を別の場所にコピーする(3)

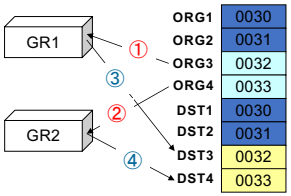
・命令の順序を変えてコピーする

【3-6行目と7-10 行目】

➢ この実行順序を変えても，ORG1～ORG4に格納されていた値を，DST1～DST4にコピーできたことになる．



3-6行目の手順



7-10行目の手順

2-3 主記憶装置の値を別の場所にコピーする(3)

・LD命令のもう１つの使い方

【11行目】　LD　GR3,GR2

➢ 11行目は，GR2に設定されている値0034_{16}をGR3に設定する［実行のようす（j）］．すなわち，汎用レジスタ間でコピーを行う命令である．

➢ 汎用レジスタ名を2つ書いた場合には，右側の汎用レジスタの値を，左側に書いた汎用レジスタに設定（コピー）する命令として機能する．

命令	書き方		命令の説明
	命令コード	オペランド	
ロード LoaD	LD	r,adr	r ← (adr) rはGR0～GR7
		r1,r2	r ← (r2) r1，r2はGR0～GR7

2-3 主記憶装置の値を別の場所にコピーする(3)

■命令の種類

命令	命令コード	説明
機械語命令 （何らかの動作をする機械語命令）	LAD RET ST など	ハードウェア(CPU)の命令に対応している． ハードウェアCOMETⅡの仕様のなかで説明されている．
アセンブラ命令 （プログラムを機械語に翻訳するアセンブラに対して指示を行う命令）	START END DS DC	アセンブラに対して指示する命令． アセンブラ言語CASLⅡの仕様のなかで説明されている．
マクロ命令 （複数の機械語命令をまとめて1つの命令で記述できるようにしたもの）	RPUSH RPOP IN OUT	いくつかの機械語命令をまとめて，1行の命令で記述できるようにした命令． アセンブラ言語で拡張された機能なので，アセンブラ言語CASLⅡの仕様のなかで説明されている．

問題[2-3]

ORG1の領域に'Z'の文字を定義しておき，これを領域（ラベル）DST1～DST4にコピーするプログラムを作成しなさい．

2-4 LD命令とLAD命令(0)

✓ プログラム**PRG0204**は，指標レジスタを用いることで，主記憶間のコピーを行っている．

```
 1 : PRG0204  START
 2 :          RPUSH
 3 :          LAD     GR1,ORG      ; ORGのアドレスをGR1へ格納
 4 :          LAD     GR2,DST      ; DSTのアドレスをGR2へ格納
 5 :          LD      GR3,3,GR1    ; ORG+3番地の値を読み込む
 6 :          LD      GR4,2,GR1
 7 :          LD      GR5,1,GR1
 8 :          LD      GR6,0,GR1
 9 :          ST      GR3,0,GR2    ; DST+0番地へGR3の値を書き込む
10 :          ST      GR4,1,GR2
11 :          ST      GR5,2,GR2
12 :          ST      GR6,3,GR2
13 :          RPOP
14 :          RET
15 : ORG      DC      '0123'
16 : DST      DS      4
17 :          END
```

3：アドレス
GR1：指標レジスタ

59

59

2-4 LD命令とLAD命令(1)

	汎用レジスタ						主記憶装置			
	GR1	GR2	GR3	GR4	GR5	GR6	DST1+0	DST2+1	DST3+2	DST4+3
(a) 3行目実行前	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(b) 3行目実行後	ORG	0000	0000	0000	0000	0000	0000	0000	0000	0000
(c) 4行目実行後	ORG	DST	0000	0000	0000	0000	0000	0000	0000	0000
(d) 5行目実行後	ORG	DST	0033	0000	0000	0000	0000	0000	0000	0000
(e) 6行目実行後	ORG	DST	0033	0032	0000	0000	0000	0000	0000	0000
(f) 7行目実行後	ORG	DST	0033	0032	0031	0000	0000	0000	0000	0000
(g) 8行目実行後	ORG	DST	0033	0032	0031	0030	0000	0000	0000	0000
(h) 9行目実行後	ORG	DST	0033	0032	0031	0030	0033	0000	0000	0000
(i) 10行目実行後	ORG	DST	0033	0032	0031	0030	0033	0032	0000	0000
(j) 11行目実行後	ORG	DST	0033	0032	0031	0030	0033	0032	0031	0000
(k) 12行目実行後	ORG	DST	0033	0032	0031	0030	0033	0032	0031	0030

60

60

2-4 LD命令とLAD命令(2)

- コピー元とコピー先のアドレスを設定

【3,4行目】　LAD　GR1,ORG

- 3行目のLAD命令は，オペランドの2番目に書いた値を，汎用レジスタに設定する命令である．
- ORGの領域に格納されている値ではなく，ORG自身，すなわちORGの領域の先頭番地がGR1に設定されることを注意しよう．
- 4行目のLAD命令で，DST領域の先頭番地がGR2に格納される．

✓ ORGの先頭番地が具体的に何番地になるかは，いろいろな条件で変わるので，具体的な数値を表すことが難しい．

✓ ここでは、ORGのまま表現している．

61

61

2-4 LD命令とLAD命令(3)

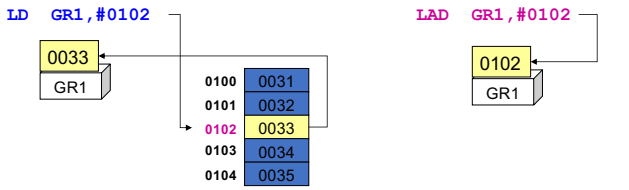
■LD命令とLAD命令の違い(1)

【a】　LD　GR1,#0102

- **LD**命令は，オペランドの0102_{16}を主記憶に対するアドレスとして，主記憶のアドレス0102_{16}番地にアクセスする．
- その番地に格納されている値を読込んで，汎用レジスタ**GR1**に設定する．

【b】　LAD　GR1,#0102

- **LAD**命令は，オペランドで指定された0102_{16}をそのまま汎用レジスタ**GR1**に設定する．
- **LAD**命令では主記憶にアクセスしないという点がポイントである．



62

62

2-4 LD命令とLAD命令(4)

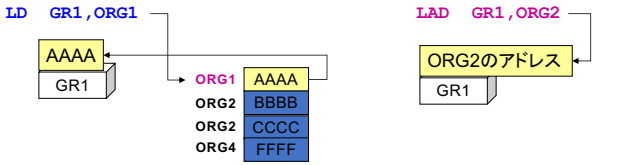
■LD命令とLAD命令の違い(2)

【a】 LD GR1,ORG1

- **LD**命令でラベルが使われていた場合には，ラベルが示す番地に記憶されている値を読み込んで**GR1**に設定する．
- **DC**命令で定数を定義した値**AAAA**$_{16}$を主記憶から読み込んで汎用レジスタ**GR1**に設定する．

【b】 LAD GR1,ORG2

- **LAD**命令は，主記憶にアクセスしないので，ラベルが使用されていても同じである．
- **ORG2**に割り当てられたアドレスを汎用レジスタ**GR1**に設定する．



63

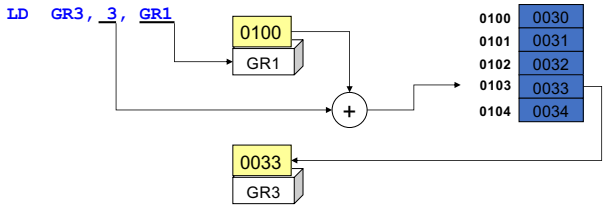
63

2-4 LD命令とLAD命令(5)

- インデックス修飾で値を読み込む(1)

【5-8行目】 LD GR3,3,GR1

- このLD命令は，インデックス修飾と呼ばれる方法により，主記憶のアドレスを決定する．
- GR1に設定されている値と第2項の3を加えた（主記憶の）番地が実際のアドレス（実効アドレス）となり，このアドレスに格納された値を読み込んで，GR3に設定する．



64

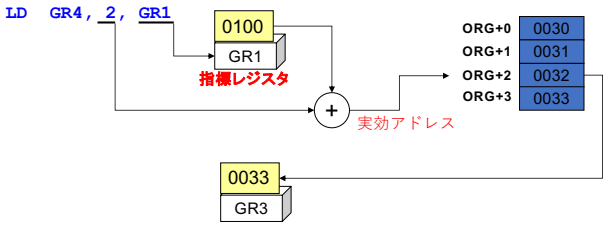
64

2-4 LD命令とLAD命令(6)

- インデックス修飾で値を読み込む(2)

【6行目】 LD GR4,2,GR1

- 3行目のLAD命令でGR1にはORG領域の先頭番地が設定されている．ORGは主記憶の番地である．
- これにオペランドの2番目の項目の2を加えたORG+2番地の値を読み込んで，GR4に設定する．



65

65

2-4 LD命令とLAD命令(7)

- インデックス修飾で値を読み込む(3)
  - このようにオペランドの第3項目の値と第2項目の値を加算して，その結果の番地にアクセスする方法をインデックス修飾アドレッシング（指標アドレス指定），
  - 加算した結果の実際にアクセスする番地を実効アドレスと呼び，このときのレジスタを指標レジスタと呼ぶ．
  - CASLⅡの場合は，指標レジスタとしてGR1～GR7しか利用できない（GR0は利用できない）約束になっているので注意が必要である．

命令	書き方		命令の説明
	命令コード	オペランド	
ロード LoaD	LD	r,adr	r ← (adr)
		r1,adr,x	r ← (adr+(x))
		r1,r2	r ← (r2)

r, r1, r2はGR0～GR7, xはGR1～GR7

66

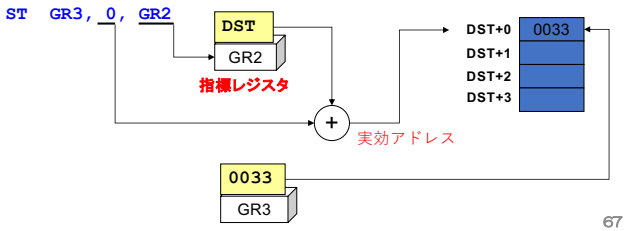
66

2-4 LD命令とLAD命令(8)

・インデックス修飾で値を書き込む(1)

【9-12行目】　ST　GR3,0,GR2

- 9行目のST命令では，オペランドの第2項目の値0と第3項目GR2の値を加算して，GR3の値を，その結果の番地に格納する．
- STとLDの命令の種類が違っても，インデックス修飾により実効アドレスを求めるまでの動作はまったく同じである．
- 最後に主記憶の値を汎用レジスタに設定するか，汎用レジスタの値を主記憶に格納するかの違いしかない．



67

67

2-4 LD命令とLAD命令(9)

・インデックス修飾で値を書き込む(2)

- 10行目のST命令：GR4の値 → DST＋1番地へ書き込み
- 11行目のST命令：GR5の値 → DST＋2番地へ書き込み
- 12行目のST命令：GR6の値 → DST＋3番地へ書き込み

- これによって，ORGに格納されていた’0123’が，DSTには’3210’として逆順にコピーされたことになる．

命令	書き方		命令の説明
	命令コード	オペランド	
ストア STore	ST	r,adr	adr ← (r)
		r1,adr,x	adr+(x) ← (r)

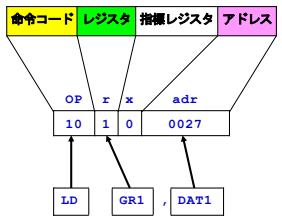
rはGR0～GR7
xはGR1～GR7

68

68

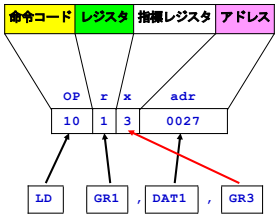
まとめ(1)

■直接アドレス指定



有効アドレス＝機械命令中のアドレス値

■インデックスアドレス指定



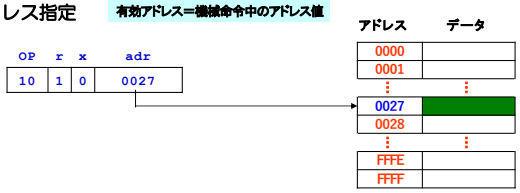
有効アドレス＝指標レジスタ＋機械命令中のアドレス値

69

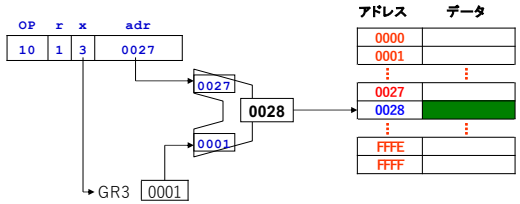
69

まとめ(2)

■直接アドレス指定　有効アドレス＝機械命令中のアドレス値



■インデックスアドレス指定　有効アドレス＝指標レジスタ＋機械命令中のアドレス値



70

70

問題[2-4]

以下の実行結果が得られるように，ST命令を使用し、プログラムを完成せよ．また，Webclassにて解答せよ

実行結果

	GR1	GR2	GR3	DAT	RST	RST+1
(a) 2行目実行後	0005	FFFF	FFFF	0005	0000	0000
(b) 3行目実行後	0005	FFFF	FFFF	0005	0005	0000
(c) 4行目実行後	0005	0001	FFFF	0005	0005	0000
(d) 5行目実行後	0005	0001	FFFF	0005	0005	0005

Program

```
1:FUKUSYU   START
2:          LD     GR1,DAT
3:          (1)
4:          LAD    GR2,1
5:          (2)
6:          RET
7:DAT       DC     5
8:RST       DS     2
9:          END
```

GR2をインデックスレジスタとして使用

2-5 キーボードからの入力する(0)

✓ キーボードから文字列を入力し，その全部または一部やそれを少し加工したものを画面に表示するプログラムである．

```
 1 : PRGO205 START
 2 :         RPUSH
 3 :         IN   DATA,LEN        ;キーボードから文字列を入力
 4 :         OUT  DATA,LEN        ;画面に出力
 5 :         OUT  DATA,LEN2       ;先頭の3文字だけ出力
 6 :         LAD  GR1,DATA
 7 :         LD   GR2,0,GR1
 8 :         LD   GR3,1,GR1
 9 :         ST   GR2,1,GR1       ;先頭の文字を2文字目に格納
10 :         ST   GR3,0,GR1       ;2文字目を先頭に格納
11 :         OUT  DATA,LEN
12 :         RPOP
13 :         RET
14 : LEN     DS   1
15 : DATA    DS   256             ;INマクロでは256語分必要
16 : LEN2    DC   3
17 :         END
```


2-5 キーボードからの入力する(1)

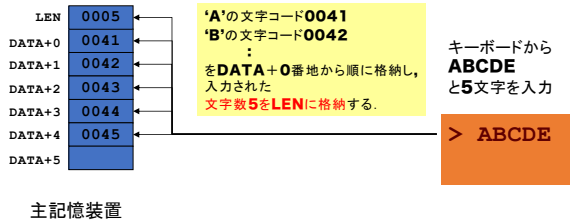
```
IN ? ABCDE      ← キーボードから入力
ABCDE           ← 4行目の出力
ABC             ← 5行目の出力
BACDE           ← 11行目の出力
```


2-5 キーボードからの入力する(2)

- IN命令

【3行目】　IN　DATA,LEN

- IN命令は，マクロ命令と呼ばれる命令で，キーボードから入力した文字を主記憶に格納する命令である．
- IN命令を実行すると，実行結果の1行目のように入力待ちの状態になる．



2-5 キーボードからの入力する(3)

・OUT命令(1)

【4行目】　OUT　DATA,LEN

- OUT命令は，オペランドで指定された領域の文字列を画面に表示する命令である．
- 下図のように，DATA番地からの領域に格納されているデータ（文字列）を，LEN番地で指定された文字数分だけ，画面に出力する．
- LENには5が格納されているので，この場合には5文字分出力される．

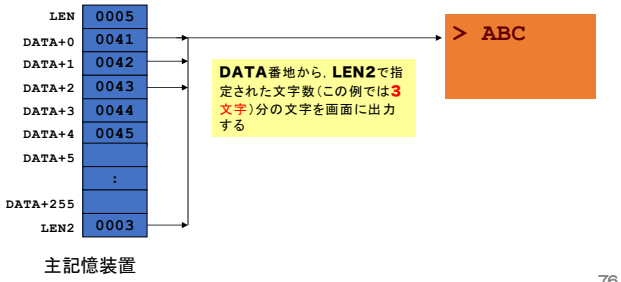
75

2-5 キーボードからの入力する(4)

・OUT命令(2)

【5,16行目】　OUT　DATA,LEN2

- LEN2は16行目で定義されていて，DC命令で3という値を定義している．
- DATAの領域には’ABCDE’と5文字格納されていたが，LEN2領域で3と指示したので3文字しか出力（表示）されない．



76

2-5 キーボードからの入力する(5)

■IN命令とOUT命令の仕様

命令	書き方		命令の説明
	命令コード	オペランド	
IN	IN	adr1,adr2	adr1　入力文字列を格納する領域のアドレス．adr1は256語分の領域を確保しておくこと． adr2　入力された文字列長を格納する領域のアドレス．
OUT	OUT	adr1,adr2	adr1　出力したい文字列を格納する領域の先頭アドレス． adr2　出力したい文字列長を格納してある領域のアドレス．

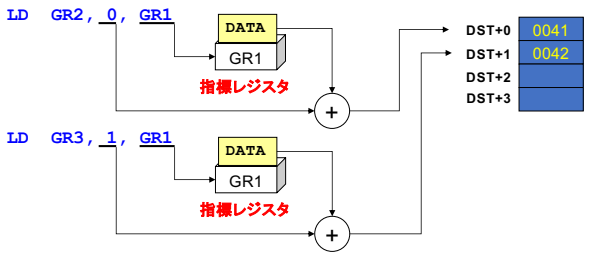
77

2-5 キーボードからの入力する(6)

・文字列の操作(1)

【6-11行目】　LD　GR2,0,GR1

- 6行目のLAD命令で，入力された文字列が格納された領域の先頭アドレスDATAをGR1に設定する．
- 7，8行目のLD命令で，インデックス修飾の方法を用いて，実効アドレスを求めて，DATA番地の値0041$_{16}$をGR2に，DATA＋1番地の値0042$_{16}$をGR3に読み込む．



78

2-5 キーボードからの入力する(7)

・ 文字列の操作(2)

【9-10行目】
```
ST   GR2,1,GR1
ST   GR3,0,GR1
```

- 9行目のST命令でGR2の値をDATA＋1番地に格納し［実行のようす（h）］.
- 10行目のST命令でGR3の値をDATA番地に格納する［実行のようす（i）］.

【11行目】
```
OUT   DATA,LEN
```

- この状態で11行目のOUT命令を実行すると，1，2文字目が入れかわった.

```
BACDE
```

79

79

問題[2-5]

キーボードから文字列を入力し，その文字列の先頭から４番目の1文字だけを画面に出力するプログラムを作成しなさい.

【出力例】
```
IN ?  RYUKYU
K
```

80

80

The END

82

82