Homework 6: Solution

Lecture, CMPEN 472

1. Getting a number from keyboard (keyboard input, numbers)

2. Printing a number on terminal window (convert each digit to ASCII characters)

Homework 6:

Lecture, CMPEN 472

1. Getting a number from keyboard (keyboard input, numbers)

2. Printing a number on terminal window (convert each digit to ASCII characters)

How to convert binary number for display, in binary, in decimal, and in hexadecimal?

How to convert ASCII representation of a number to a binary number?

Time 20210301

Lecture, CMPEN 472

Homework 6:

How to convert binary number for display, in binary, in decimal, and in hexadecimal?

How to convert ASCII representation of a number to a binary number?

Time 20210301

Homework 6:

Lecture, CMPEN 472

How to convert binary number for display, in binary, in decimal, or in hexadecimal?

1. Separate each digit – by dividing the number by 2, 10, or 16 and collecting the remainder
2. Convert the remainders to ASCII characters by adding ASCII bias
3. Then print the characters in reverse order (MSB first)

How to convert ASCII representation of a number to a binary number?

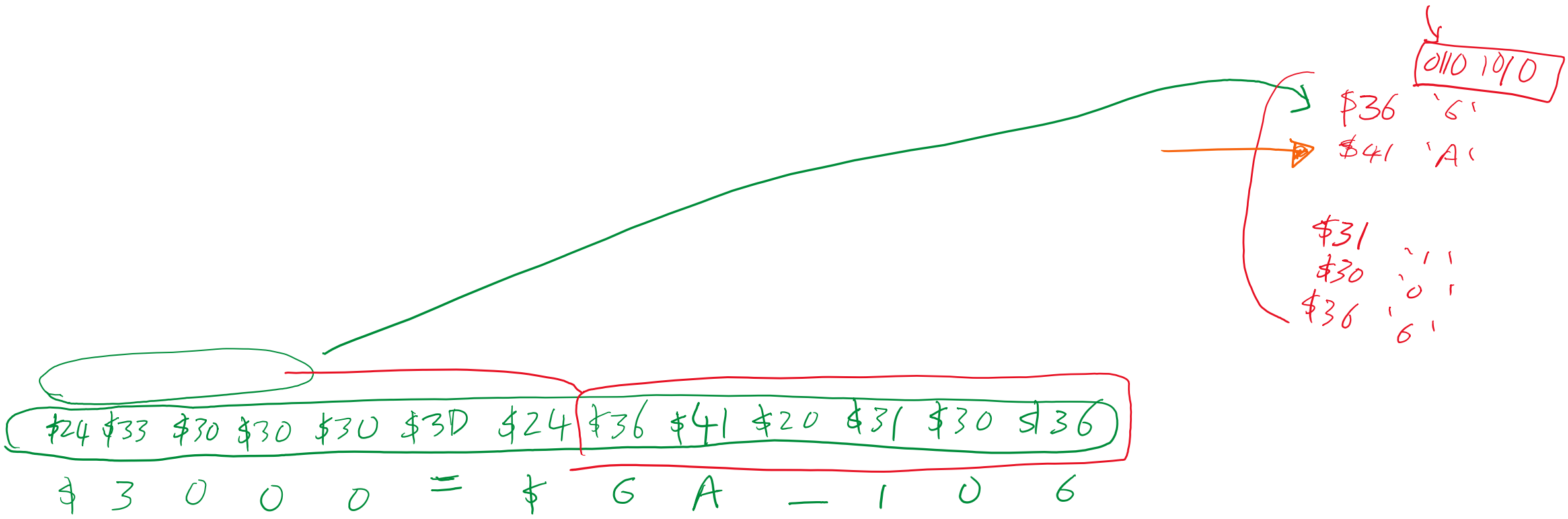
1. Take each character and subtract ASCII bias
2. Take each digit and multiply its weight:
   a. 1,2,4,8,16,32,… for binary
   b. 1,10,100,1000,10000,… for decimal
   c. 1,16,256,4096, 65536,… for hexadecimal
3. Add all of the weighted numbers

Time 20210301

Homework 6: Number Printing

Lecture, CMPEN 472

Example of printing $6A from the content of accumulator A (hexadecimal print)



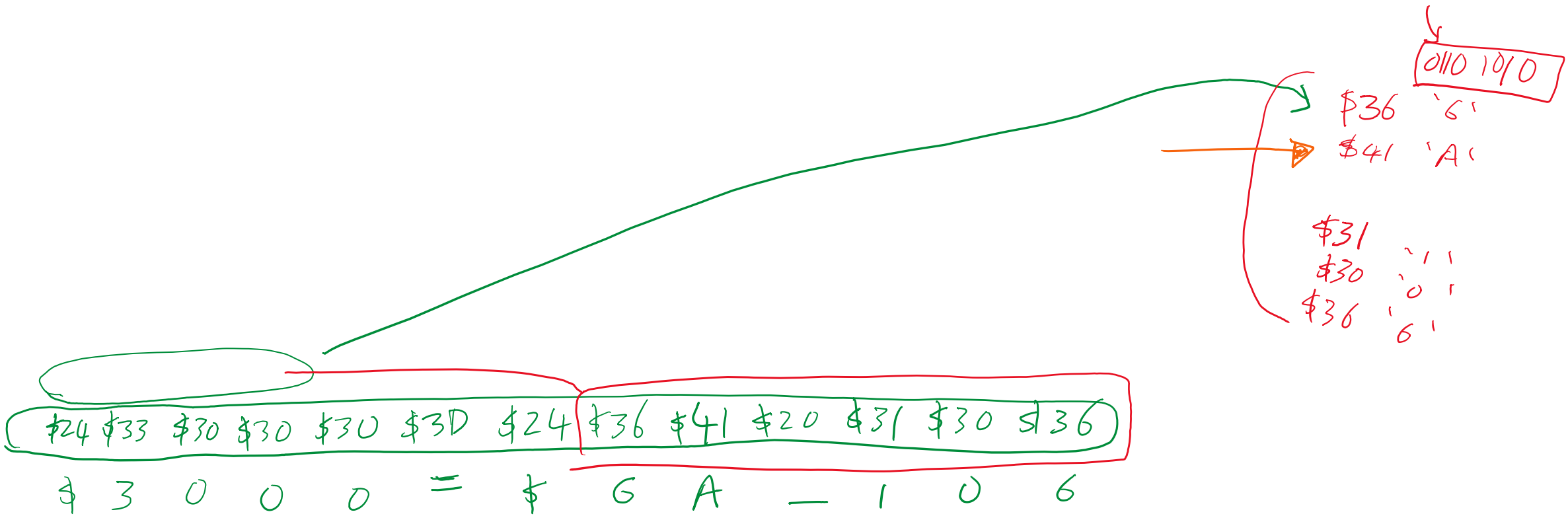
Time 20210229

Homework 6:

Lecture, CMPEN 472

Example of printing $6A from the content of accumulator A (hexadecimal print)

accumulator A contains number %01101010=$6A=106



Time 20210229

Objective

To learn how to use arithmetic instructions and write basic system I/O subroutines. Learn user interface through a serial port.

Instruction

1. Write a user-friendly system program for the following commands:

   S: Show the contents of memory location
   W: Write the data byte to memory location
   QUIT: Quit the main program, run 'Type writer' program.

2. **Command S:** This command shows the contents of memory location specified by the address in hexadecimal number followed by the 'S' character. For example, if the data $6A is stored in memory location $3000, a user types in the first line ending with Enter/Return key and the following should be displayed on the HyperTerminal connected to the HCS12 board:

   >S$3000
   $3000 = $6A 106
   >

   The data $6A is printed in **both hexadecimal and decimal** number format. The character '>' is the prompt for this program.

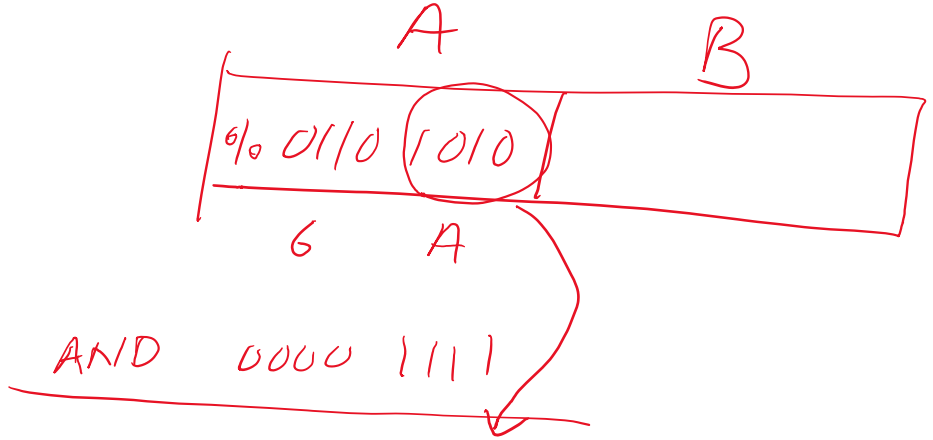
3. **Command W:** this command writes data into the memory location specified by the address in hexadecimal followed by the 'W' character. The data to be written to the memory location is followed by a space and it can be specified by hexadecimal '$' or just decimal number. For example, if one wants to store the data $6A in memory location $3001, a user types in the first line ending with Enter/Return key and the following should be displayed on the HyperTerminal connected to the HCS12 board:

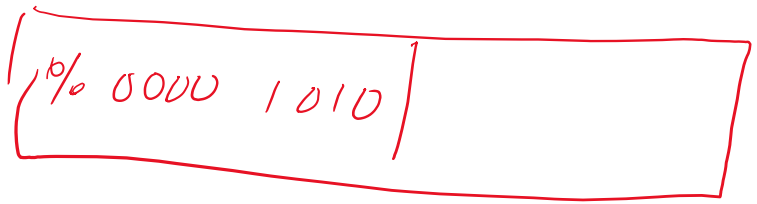
Lecture, CMPEN 472

Homework 6: coding

```
      LDAA  $3000
      ANDA  #$0F
      CMPA  #9
      BHi   Alpha
      ADDA  #$30
      STAA  $3004
      BRA   next2
Alpha
      ADAA  #$37
      STAA  $3004
      BRA   next2
```

'6'
'A'





$0A
$35
$3F

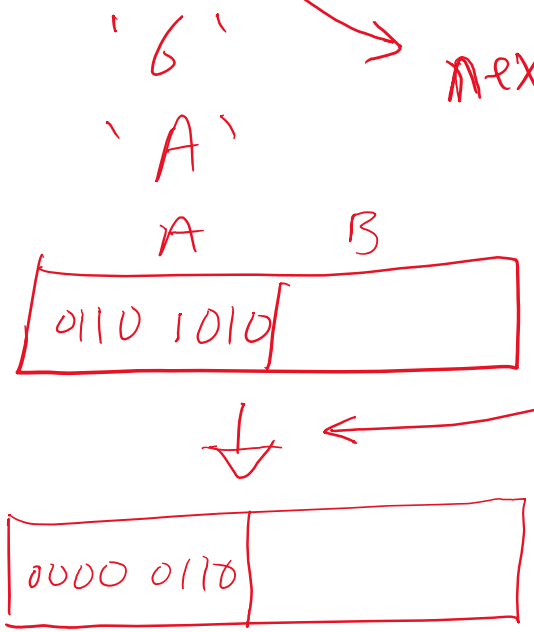
$0A
+ $
$41 → 'A'

Time 20210301

Lecture, CMPEN 472

Homework 6:

```
LDAA   $3000
ANDA   #$0F
CMPA   #9
BHi    Alpha
ADDA   #$30
STAA   $3004
BRA    next2
Alpha
ADAA   #$37
STAA   $3004
BRA    next2
```



next2

```
LDAA   $3000
LSRA
LSRA
LSRA
LSRA
CMPA   #9
BHi    Alpha2
ADDA   #$30
STAA   $3003
BRA    next3
```

Alpha2

```
ADDA   #$37
STAA   $3003
BRA    next3
```

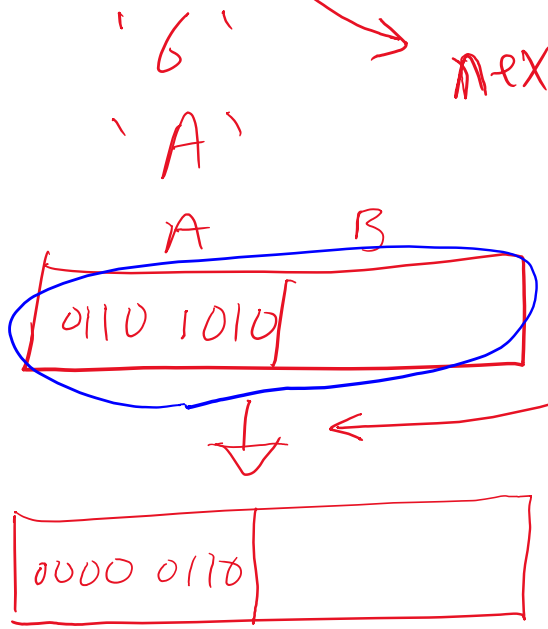
DONE hex

Time 20210301

Lecture, CMPEN 472

Homework 6:

```
LDAA  $3000
ANDA  #$0F
CMPA  #9
BHI   Alpha
ADDA  #$30
STAA  $3004
BRA   next2
Alpha
ADAA  #$37
STAA  $3004
BRA   next2
```



For 16 bit number, do the same for B register.

```
LDAA  $3000
LSRA
LSRA
LSRA
LSRA
CMPA  #9
BHI   Alpha2
ADDA  #$30
STAA  $3003
BRA   next3
Alpha2
ADDA  #$37
STAA  $3003
BRA   next3
DONE  hex
```

Time 20210301

Homework 6:

Lecture, CMPEN 472

Example of printing 106 from the content of accumulator A (decimal print)

accumulator A contains number %01101010=$6A=106

Homework 6:

Lecture, CMPEN 472

Example of printing 106 from the content of accumulator A (decimal print)

accumulator A contains number %01101010=$6A=106

Keep dividing by 10 the number in accumulator A to separate the decimal digits:

If you collect the remainders after dividing by 10, you get the numbers 1, 0, and 6.

And you may convert them to ‘1’, ‘0’, and ‘6’ by adding the ASCII bias $30.

Homework 6:

Lecture, CMPEN 472

0110 1010 ÷ 0000 1010 = 0000 1010 ... 0000 0110

$6A = 106 $0A = 10 $0A = 10 $06 = 6

0000 1010 ÷ 0000 1010 = 0000 0001 ... 0000 0000

$0A = 10 $0A = 10 $01 = 1 $00 = 0

0000 0001 ÷ 0000 1010 = 0000 0000 ... 0000 0001

$01 = 1 $0A = 10 $00 = 0 $01 = 1

Stop

123 ÷ 10 = 12 ... 3

12 ÷ 10 = 1 ... 2

1 ÷ 10 = 0 ... 1

Time 20210301

Homework 6: Code

Lecture, CMPEN 472

```
; getting 3 decimal digits from accumulator A data
decprint
        TAB
        CLRA
        LDX     #10
        IDIV
        STAB    $3004
        TFR     X,D
        LDX     #10
        IDIV
        STAB    $3005
        TFR     X,D
        LDX     #10
        IDIV
        STAB    $3006
; now all 3 decimal digits are done
        BRA     nx
nx:
        LDAA    #$30
        ADDA    $3006
        JSR     putchar
        LDAA    #$30
        ADDA    $3005
        JSR     putchar
        LDAA    #$30
        ADDA    $3004
        JSR     putchar
; now all 3 decimal digits are printed
        RTS
```

Time 20210301

Homework 6:

Lecture, CMPEN 472

Time 20210301

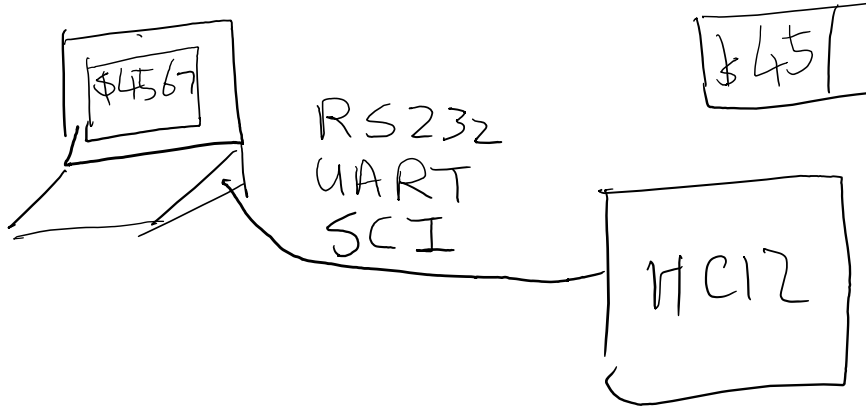
Lecture, CMPEN 472

Homework 6:

How to convert binary number for display, in binary, in decimal, and in hexadecimal

How to convert ASCII representation of a number to a binary number?

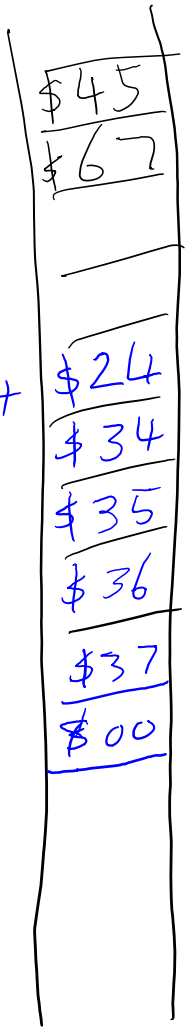
$4567



A B

$45

$3000



LDX #$3004
JSR printmsg

Question

Take $4567,
then produce
$24, $34, $35, $36, $37, $00
↓ ↓
'$' How?

$45 = 'E'
$67 = 'g'

LDAA $3000
JSR putchar

LDAA $3001
JSR putchar

Time 20210301

Lecture, CMPEN 472

X D X B

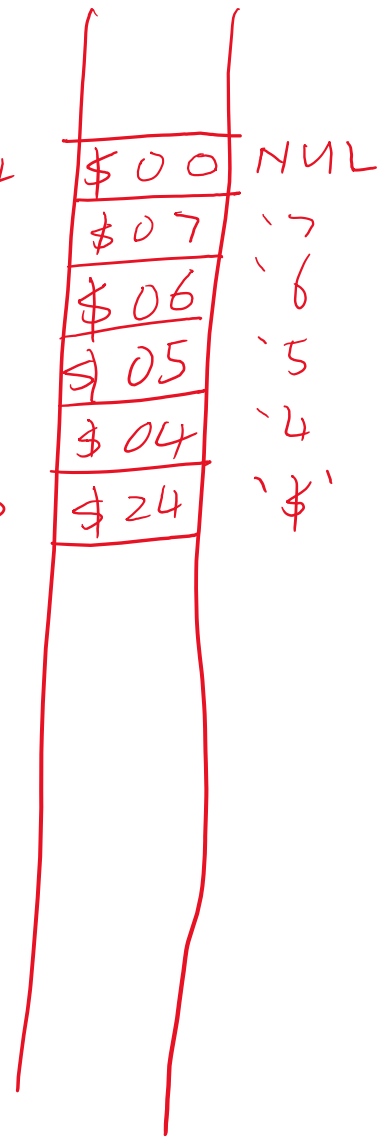
Lectures:

$4567 ÷ 16 = $0456 ... 7

$0457 ÷ 16 = $0045 ... 6

$0045 ÷ $10 = $0004 ... 5

$0004 ÷ $10 = $0000 ... 4



Time 20210303

Lecture, CMPEN 472

X D X B

Lectures:

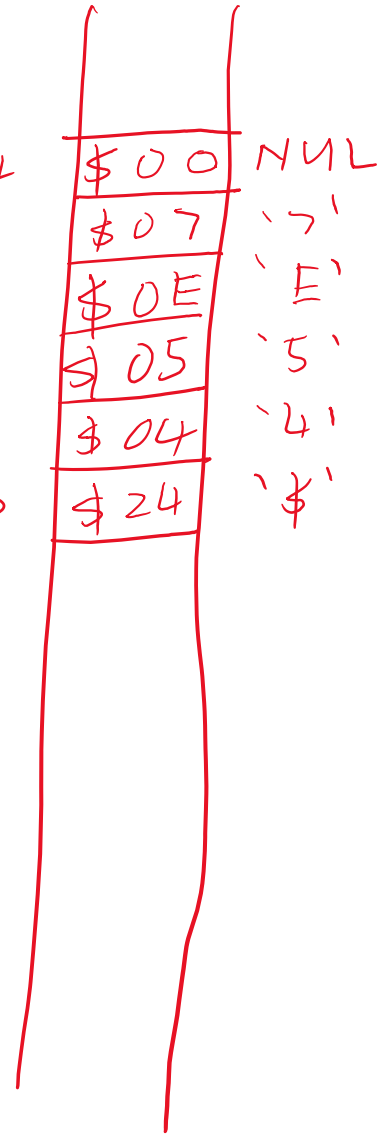
$45E7 ÷ 16 = $045E ... 7

$0457 ÷ 16 = $0045 ... E

$0045 ÷ $10 = $0004 ... 5

$0004 ÷ $10 = $0000 ... 4

$3004



$3E ⇒ '

$45 ⇒ 'E'

Time 20210303

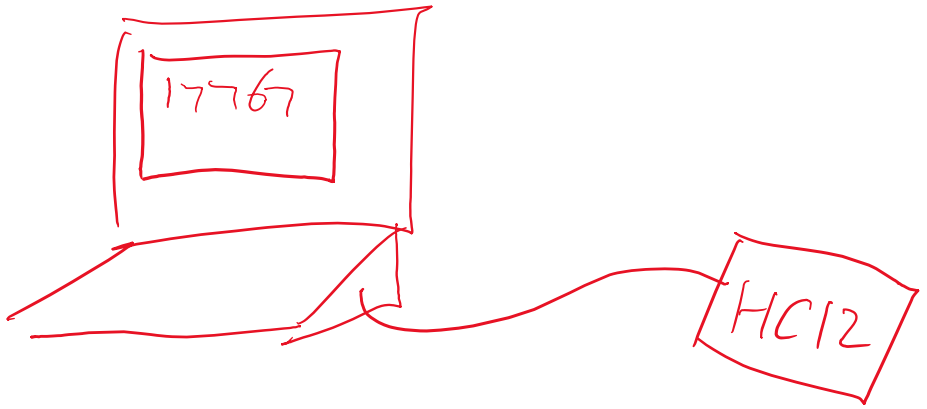
Lecture, CMPEN 472

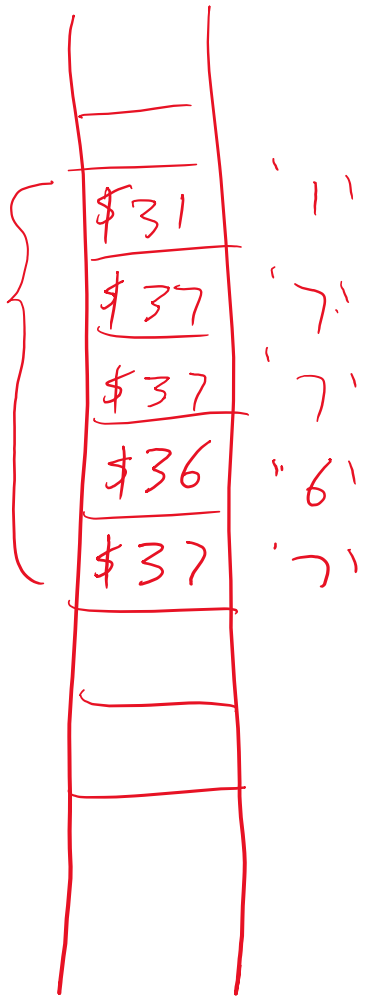
Lectures:

$4567 = 17767_d$

$= \%0100\ 0101\ 0110\ 0111$

4 5 6 7





Time 20210303

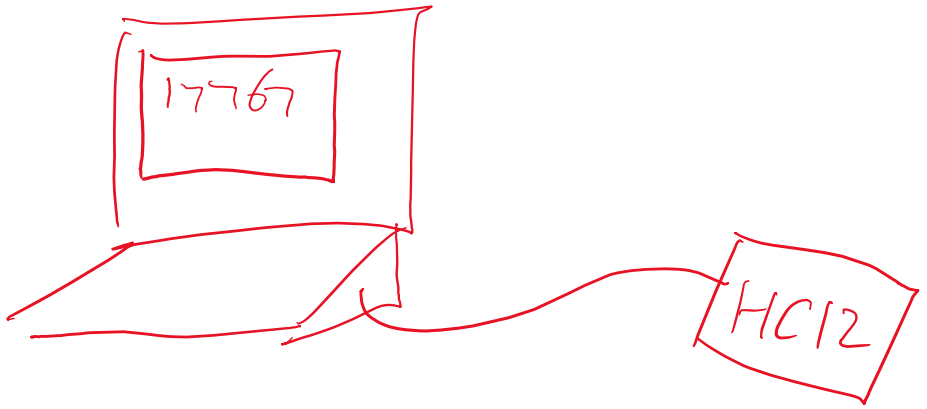
Lecture, CMPEN 472

Lectures:

$4567 = 17767_d$

$= \%0100\ 0101\ 0110\ 0111$

4 5 6 7



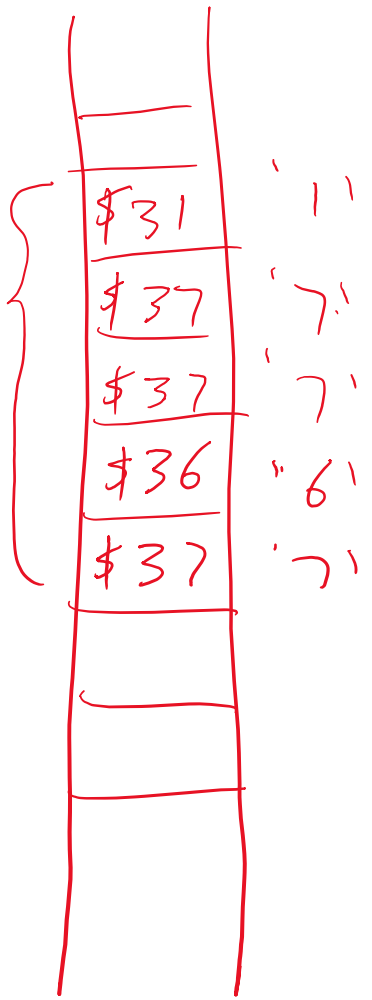
$4567 \div 10_d = \$06F0 \ldots 7$

$\$06F0 \div \$A = \$00B1 \ldots 6$

$\$00B1 \div \$A = \$0011 \ldots 7$

$\$0011 \div \$A = \$0001 \ldots 7$

$\$0001 \div \$A = \$0000 \ldots 1$



Time 20210303

Lecture, CMPEN 472

Homework 6:

Questions – number (decimal or hexadecimal) entering (keyboard)
– number (decimal or hexadecimal) display (printing on screen)

Homework 6: Keyboard Input Number

Lecture, CMPEN 472

How to convert binary number for display, in binary, in decimal, or in hexadecimal?

How to convert ASCII representation of a number to a binary number?

1. Take each character and subtract ASCII bias
2. Take each digit and multiply its weight:
   a. 1,2,4,8,16,32,… for binary
   b. 1,10,100,1000,10000,… for decimal
   c. 1,16,256,4096, 65536,… for hexadecimal
3. Add all of the weighted numbers

Time 20210301

File Edit View History Bookmarks Tools Help

CMPEN 472, Spring 2021, Penn | CMPEN 472 Homework 6, The

www.cse.psu.edu/~kxc104/class/cmpen472/21s/hw/hw6/hw6.html 110%

ZoomPSU ZoomMM ZoomUS Canvas Outlook Kyusun Google Portal PSU per Bible Adobe DDGo Other Bookmarks

QUIT: Quit the main program, run 'Type writer' program.

2. **Command S:** This command shows the contents of memory location specified by the address in hexadecimal number followed by the 'S' character. For example, if the data $6A is stored in memory location $3000, a user types in the first line ending with Enter/Return key and the following should be displayed on the HyperTerminal connected to the HCS12 board:

```
>S$3000
  $3000 = $6A    106
>
```

Quiz 20210303:

The data $6A is printed in **both hexadecimal and decimal** number format. The character '>' is the prompt for this program.

3. **Command W:** this command writes data into the memory location specified by the address in hexadecimal followed by the 'W' character. The data to be written to the memory location is followed by a space and it can be specified by hexadecimal '$' or just decimal number. For example, if one wants to store the data $6A in memory location $3001, a user types in the first line ending with Enter/Return key and the following should be displayed on the HyperTerminal connected to the HCS12 board:

```
>W$3001 106
  $3001 = $6A    106
>
```

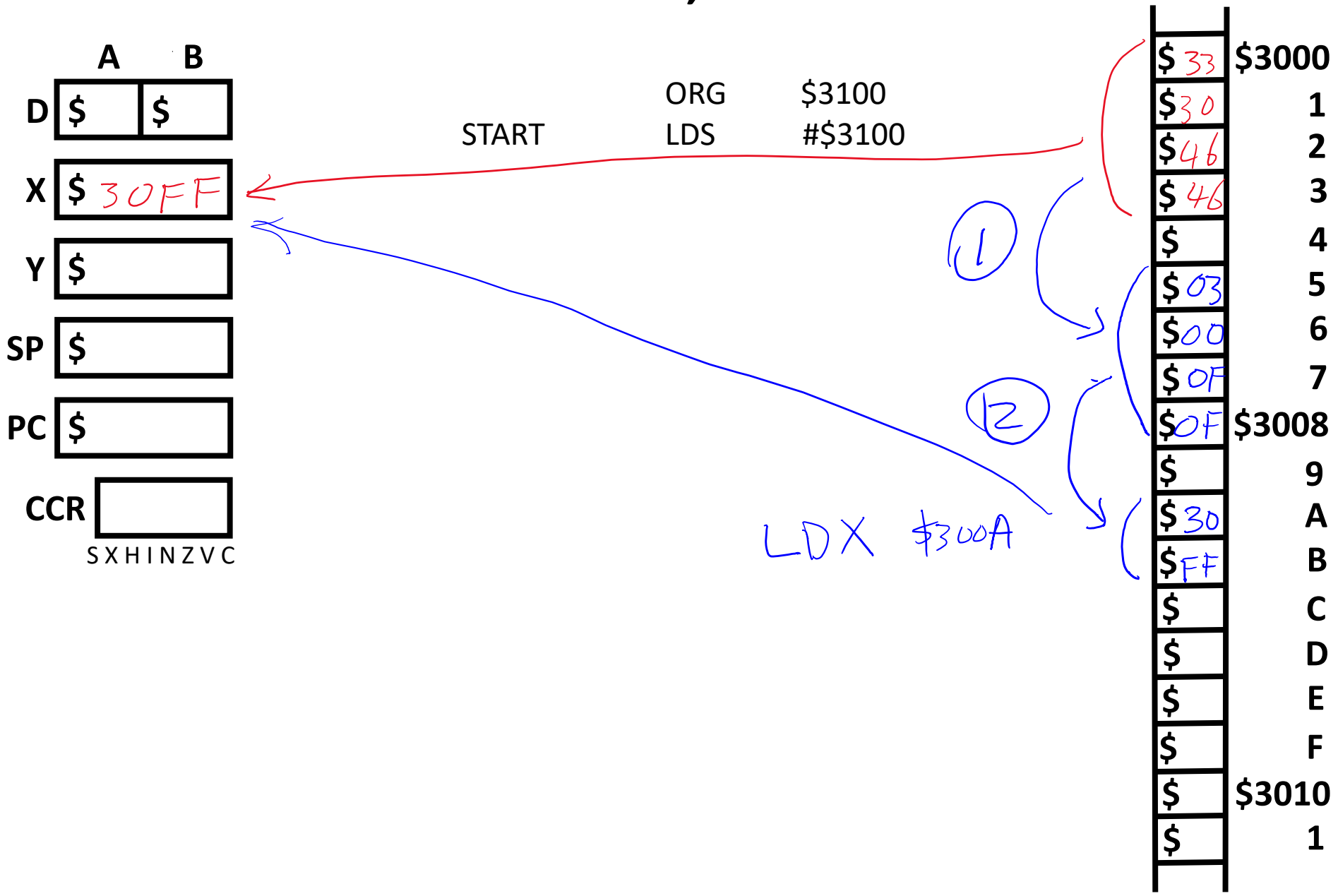
Or

```
>W$3001 $6A
  $3001 = $6A    106
>
```

As a result, the data $6A is stored in the memory location $3001 and it is shown with the 'S' command. The 'W' command accepts both '106' or '$6A' as an 8 bit number.

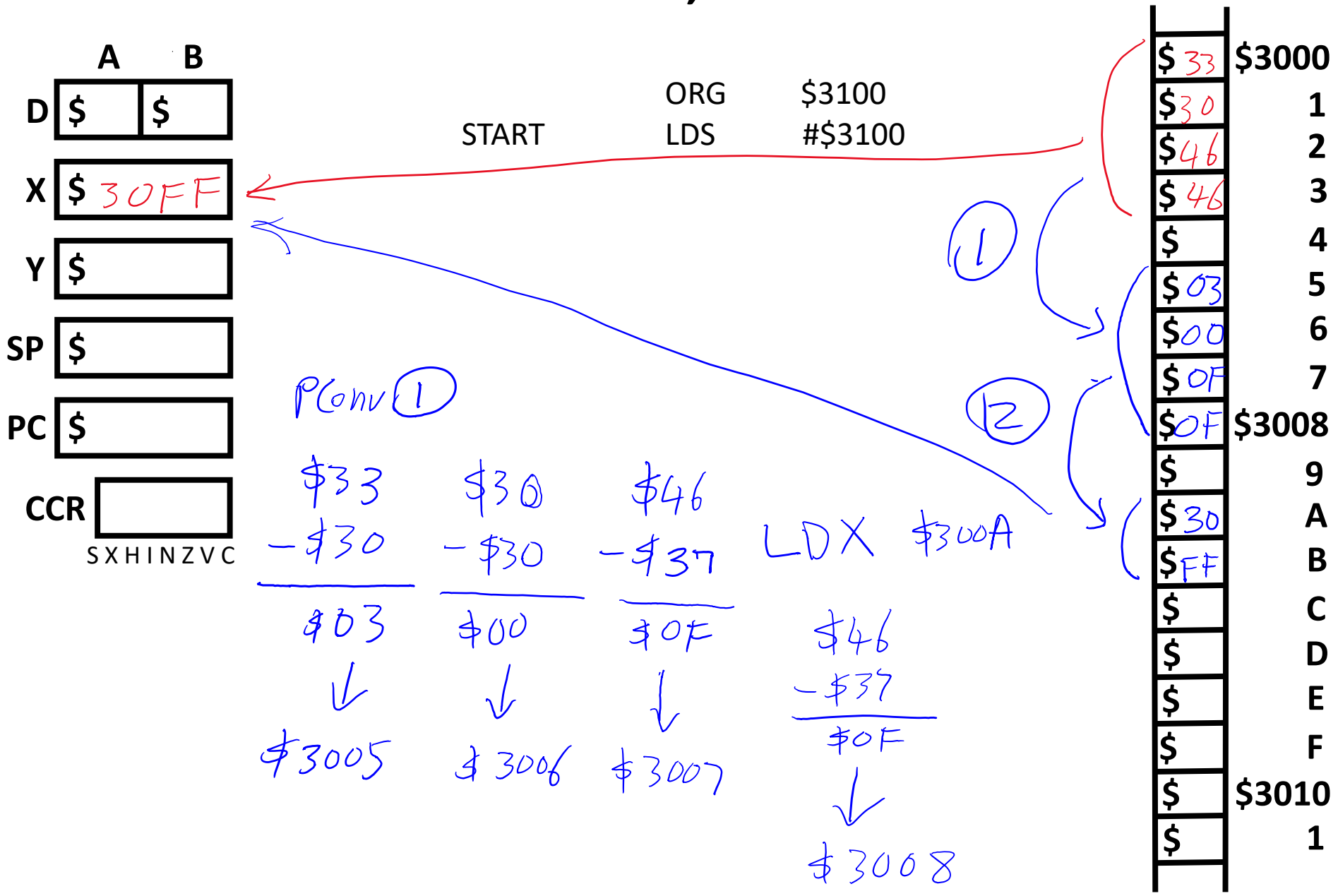
Time 20210303

Lecture, CMPEN 472



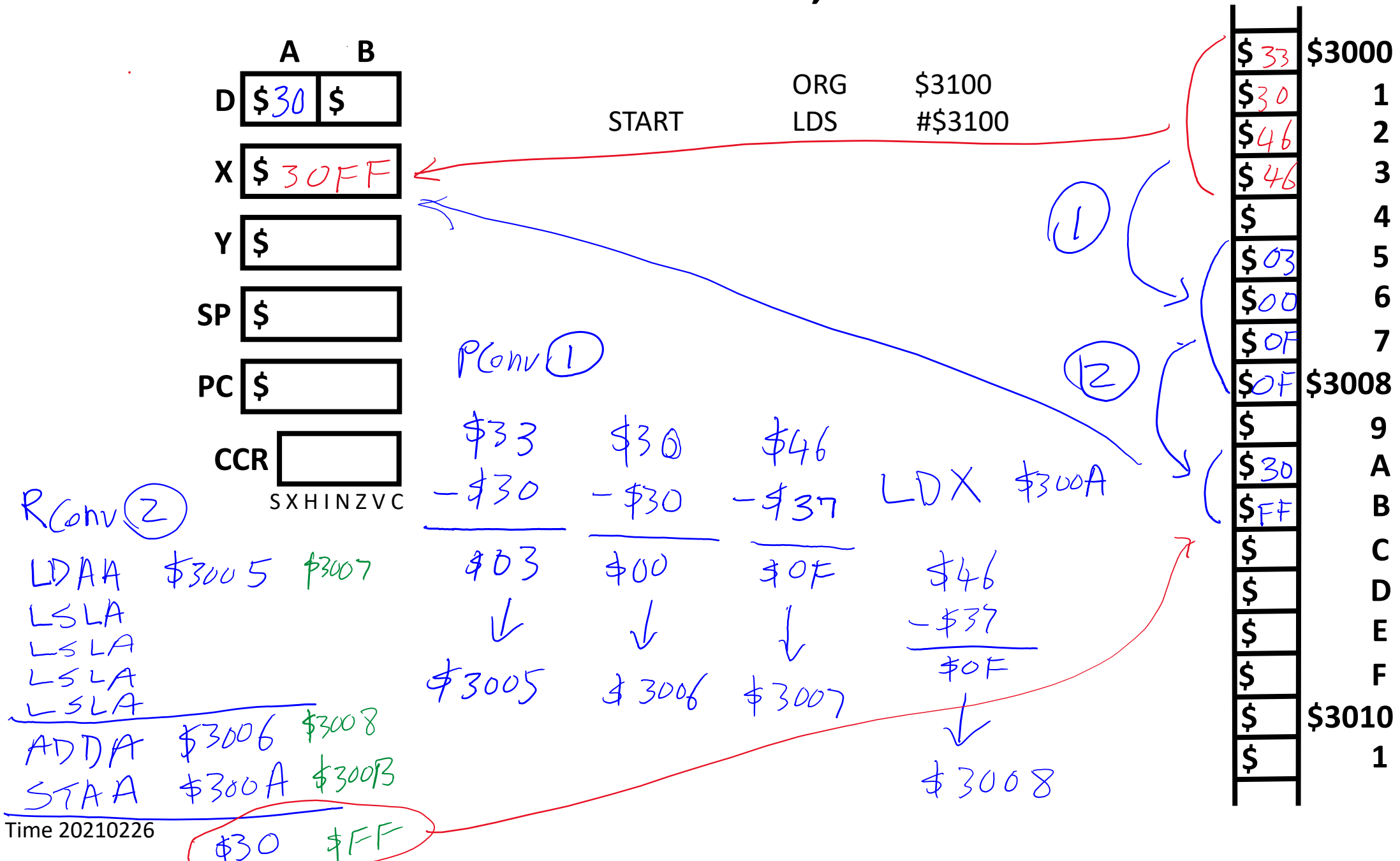
Time 20210226

Lecture, CMPEN 472



Time 20210226

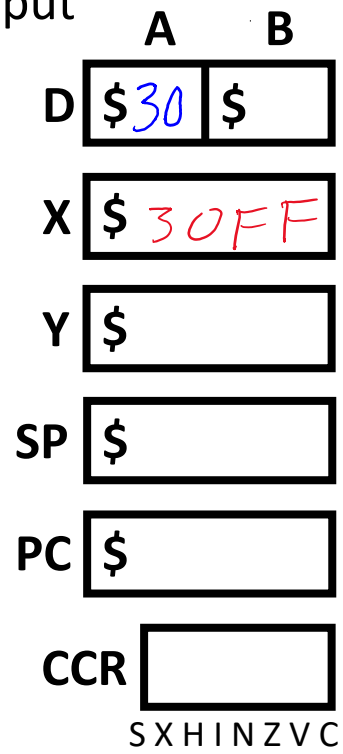
Lecture, CMPEN 472



Homework 6: Code

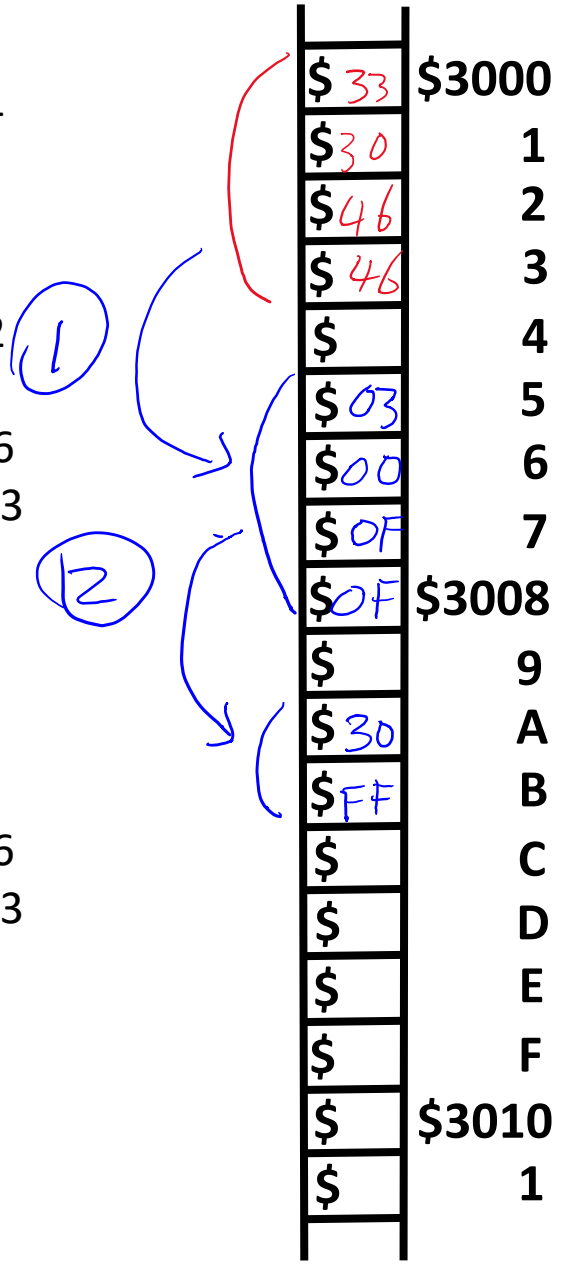
Lecture, CMPEN 472

Keyboard Input




```
keyin1  LDAA   $3000
        CMPA   #$30
        BLO    err
        CMPA   #$39
        BHI    alph1
        SUBA   #$30
        STAA   $3005
        BRA    keyin2
Alph1   CMPA   #$41
        BLO    err
        CMPA   #$46
        BHI    err
        SUBA   #$37
        STAA   $3005
        BRA    keyin2
```

```
keyin2  LDAA   $3001
        CMPA   #$30
        BLO    err
        CMPA   #$39
        BHI    alph2
        SUBA   #$30
        STAA   $3006
        BRA    keyin3
Alph2   CMPA   #$41
        BLO    err
        CMPA   #$46
        BHI    err
        SUBA   #$37
        STAA   $3006
        BRA    keyin3
```

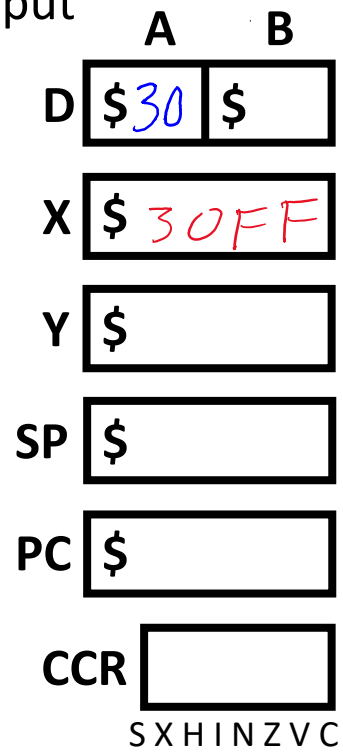


Time 20210226

Homework 6: Code

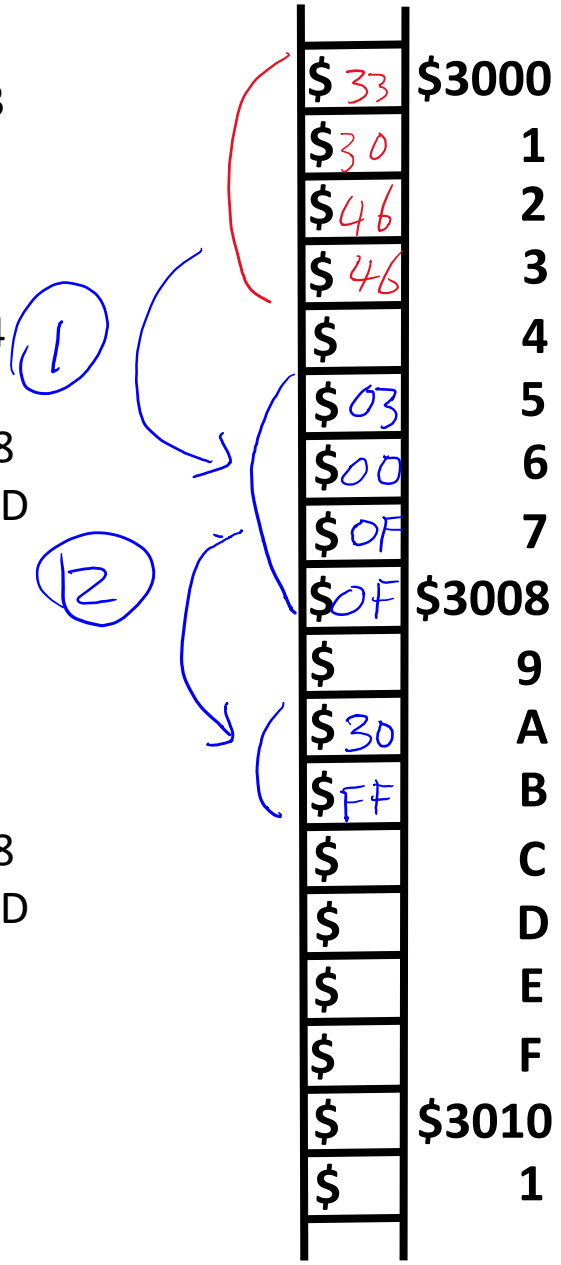
Lecture, CMPEN 472

Keyboard Input




```
keyin3  LDAA   $3002
        CMPA   #$30
        BLO    err
        CMPA   #$39
        BHI    alph3
        SUBA   #$30
        STAA   $3007
        BRA    keyin4
Alph3   CMPA   #$41
        BLO    err
        CMPA   #$46
        BHI    err
        SUBA   #$37
        STAA   $3007
        BRA    keyin4
```

```
keyin4  LDAA   $3003
        CMPA   #$30
        BLO    err
        CMPA   #$39
        BHI    alph4
        SUBA   #$30
        STAA   $3008
        BRA    keyinD
Alph4   CMPA   #$41
        BLO    err
        CMPA   #$46
        BHI    err
        SUBA   #$37
        STAA   $3008
        BRA    keyinD
```



Time 20210226

Homework 6: Code

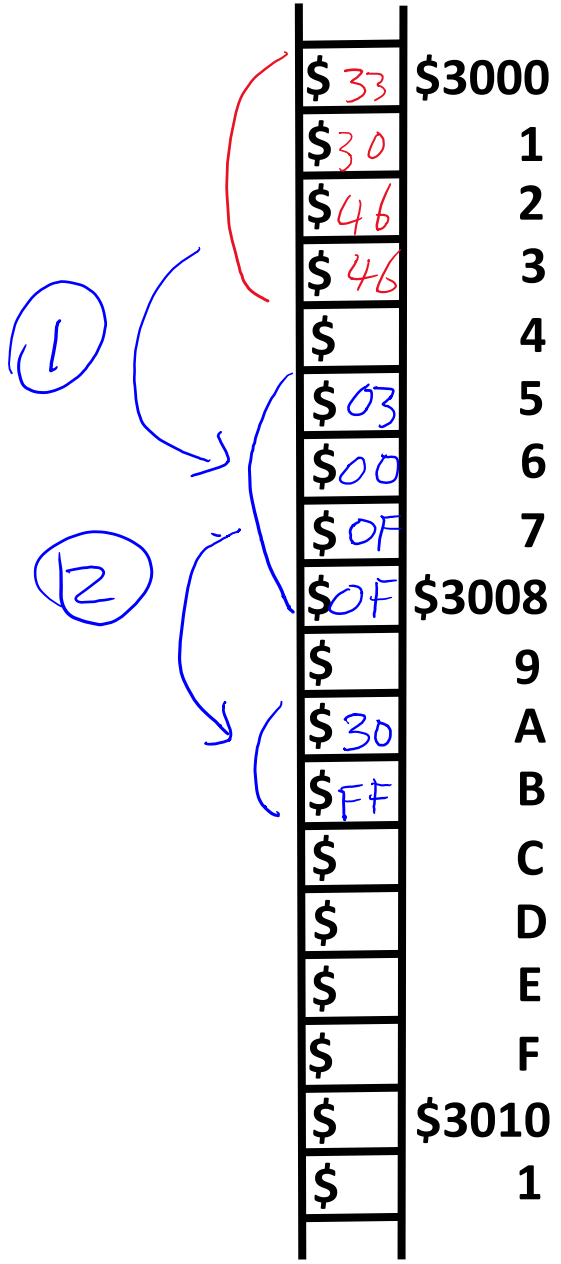
Lecture, CMPEN 472

Keyboard Input

	A	B
D	$30	$
X	$ 30FF	
Y	$	
SP	$	
PC	$	
CCR		

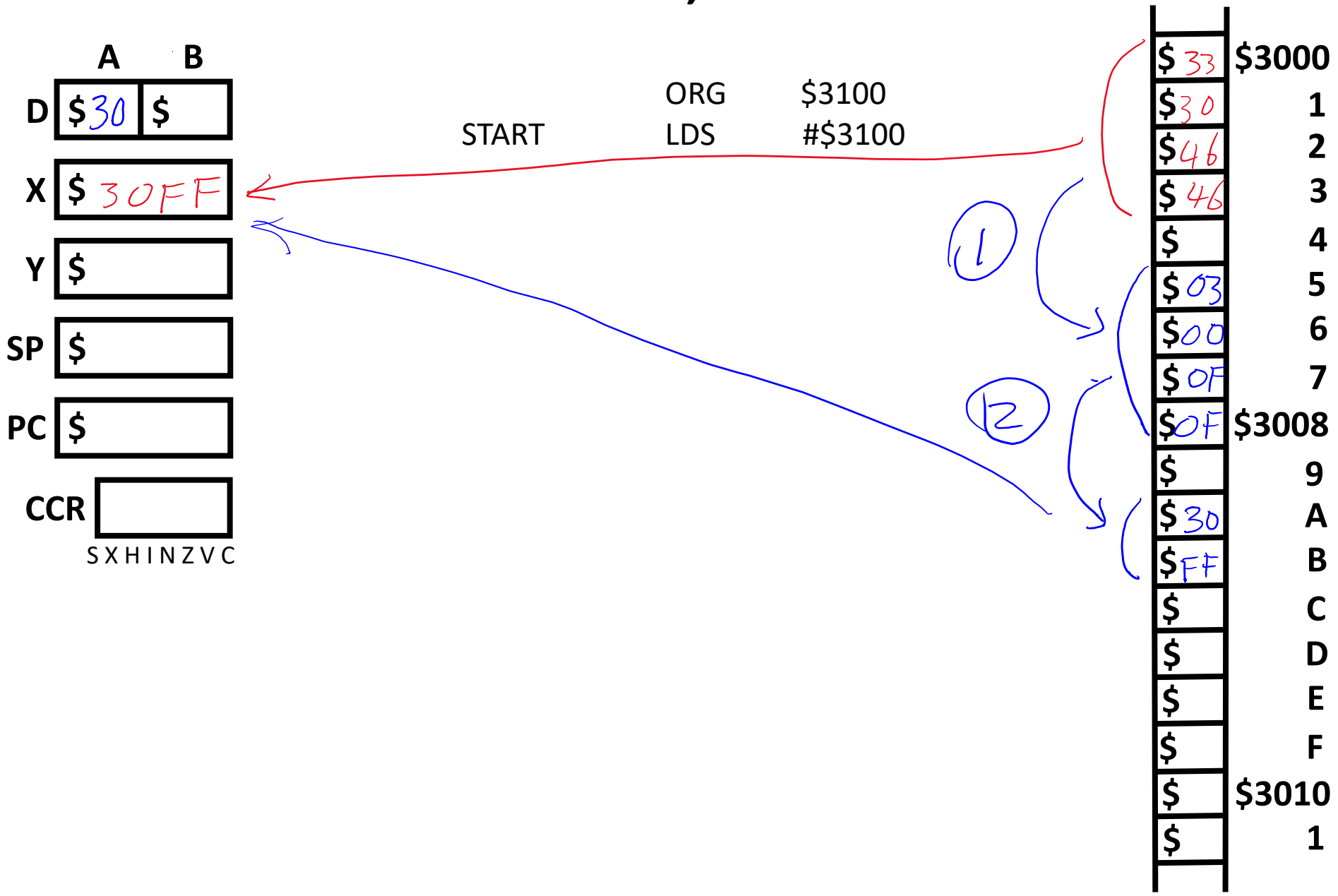
S X H I N Z V C

```
keyinD  LDAA   $3005
        LSLA
        LSLA
        LSLA
        LSLA
        ADDA   $3006
        STAA   $300A
        LDAA   $3007
        LSLA
        LSLA
        LSLA
        LSLA
        ADDA   $3008
        STAA   $300B
        LDX    $300A
        RTS
```



Time 20210226

Lecture, CMPEN 472



Time 20210226

ASCII Table and Description

ASCII stands for American Standard Code for Information Interchange. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort. ASCII was developed a long time ago and now the non-printing characters are rarely used for their original purpose. Below is the ASCII character table and this includes descriptions of the first 32 non-printing characters. ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure. If someone says they want your CV however in ASCII format, all this means is they want 'plain' text with no formatting such as tabs, bold or underscoring - the raw format that any computer can understand. This is usually so they can easily import the file into their own applications without issues. Notepad.exe creates ASCII text, or in MS Word you can save a file as 'text only'

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

Time 20210303