

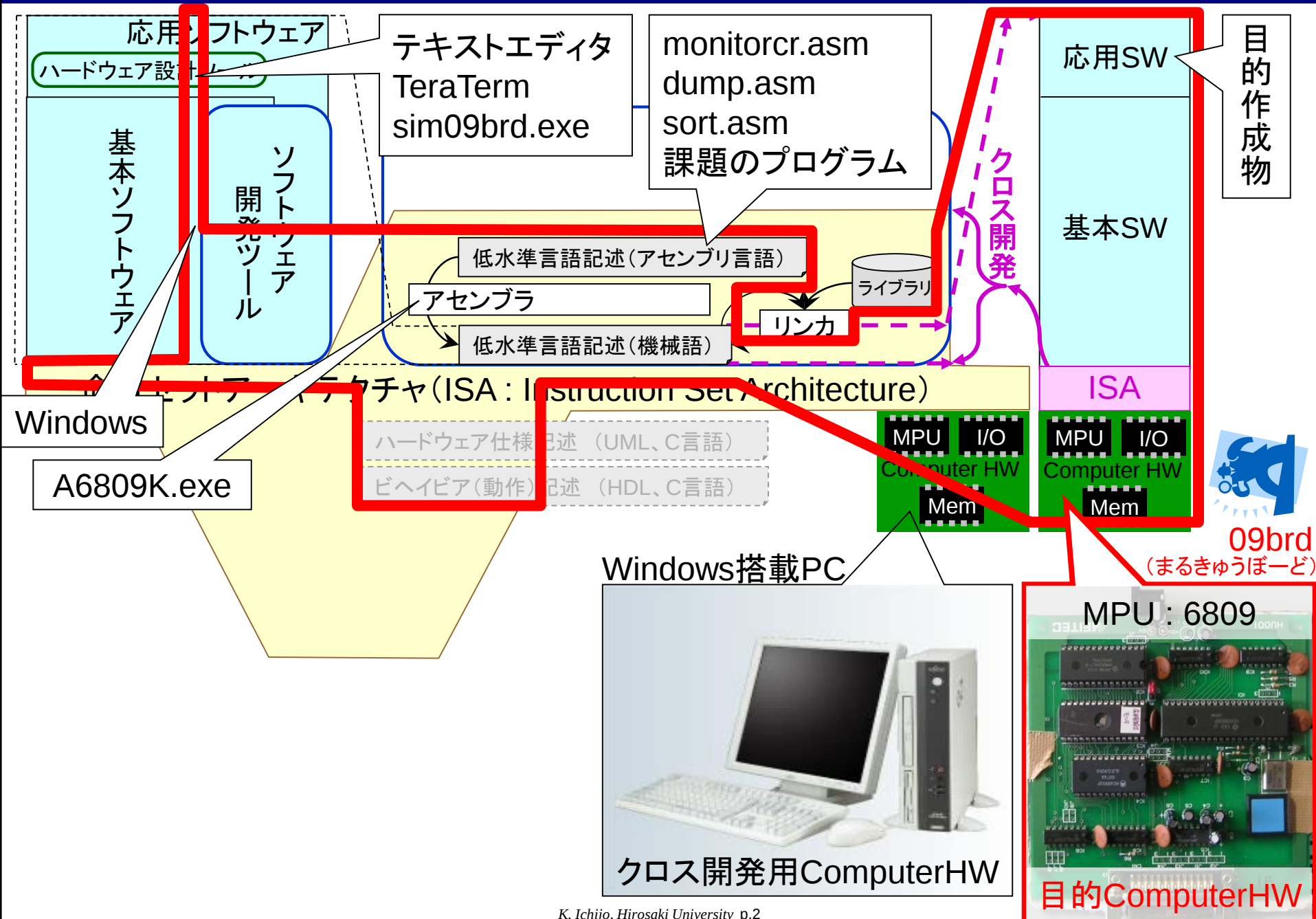
組み込みプロセッサ、DSPの利用例

KORG シンセサイザー RADIAS	TI TMS320VC5502(C55x)
ONKYO、YAMAHA AVアンプ	TI TMS320DA7xx(C67x)
NEC W-CDMA基地局	TI TMS320TCI6487(C64x)
各社 携帯電話 (NTT DoCoMo FOMAなど)	TI、ARM OMAP/OMAP2 (TI DSP+ARM)
Palm PDA Tungsten T	TI、ARM OMAP1510(C55x+ARM9)
任天堂 GAMEBOY ADVANCE	ARM ARM7
任天堂 ニンテンドーDS	ARM ARM9+ARM7
Apple iPod	ARM ARM7
Apple PDA Newton	ARM ARM7
Psion PDA Series5	ARM ARM7
Corega、PLANEX ブロードバンドルータ	ARM ARM9
SEGA ドリームキャスト	Renesas SH-4
産総研認定ベンチャー ヒューマノイドロボット HRP-2m Choromet	Renesas SH-4
HITACHI WinCE端末 PERSONA	Renesas SH-4
IOデータ NAS LAN-iCN2、LANDISK	Renesas SH-4
アイコム、マイクロ総合研究所 ブロードバンドルータ	Renesas SH-4
SANYO カーナビ NV-HD870DT	Renesas SH-4
Clarion 業務用車両向け汎用車載端末 CA-7000	Renesas SH-4

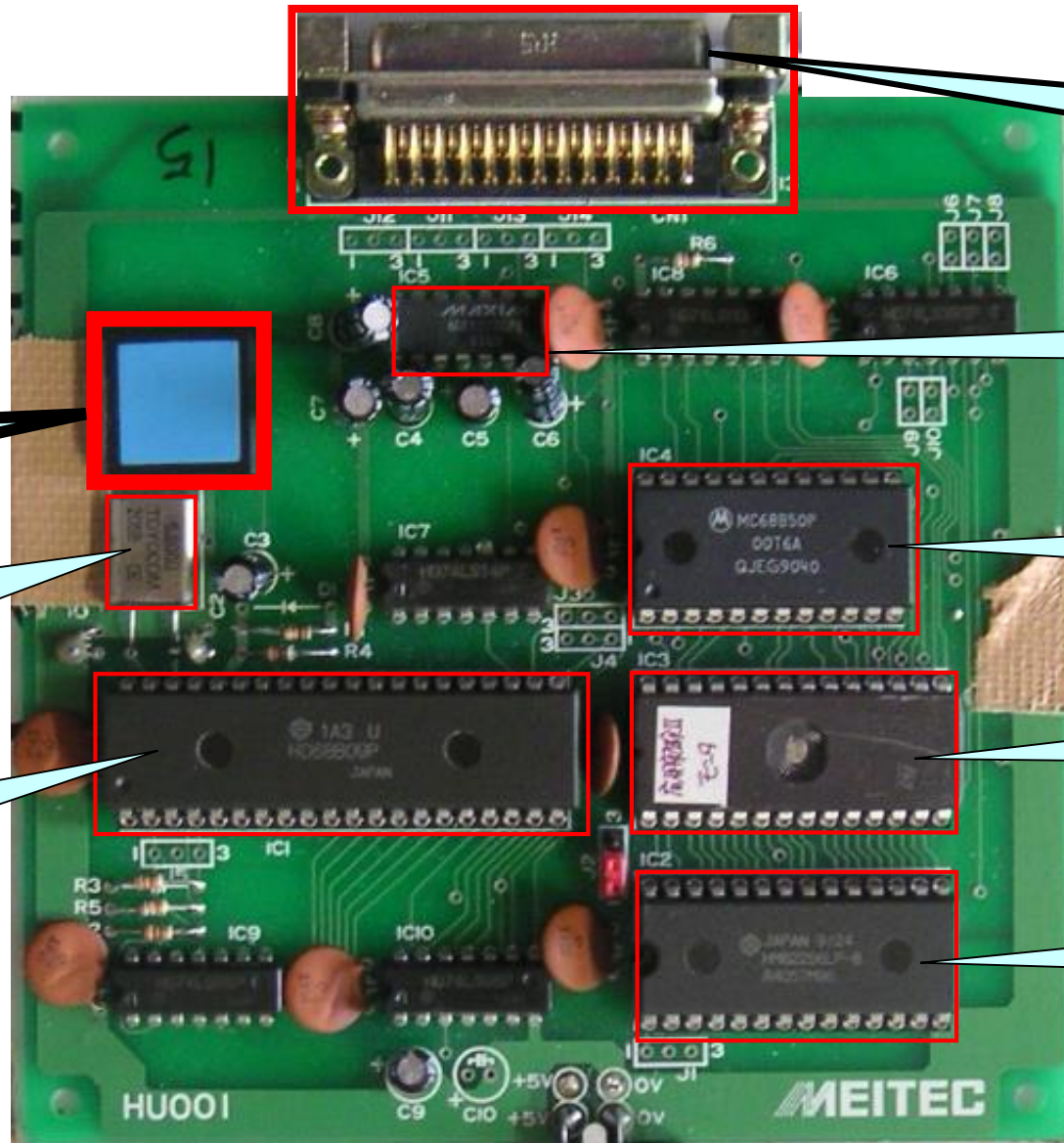
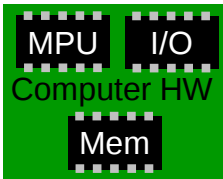
↑青文字がDSP (Digital Signal Processor)
プロセッサの一形態

「パソコン」以外のコンピュータシステムも沢山ある。

目的コンピュータハードウェア 09brd(まるきゅうボード)



09brd(まるきゅうボード)の概要 (1/2)



I/O装置

シリアルポート
コネクタ

Driver IC
MAX232

ACIA IC
MC68B50P

ROM IC
27256

RAM IC
67256

主記憶装置

リセットボタン

水晶発信器
(クロック信号
発生器)

MPU IC
HD68B09P

CPU(プロセッサ)

電源コネクタ
(電源On/Off)

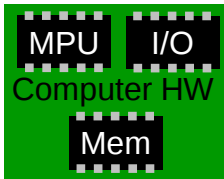
【注】ACIA:

Asynchronous Communications Interface Adapter

■特徴

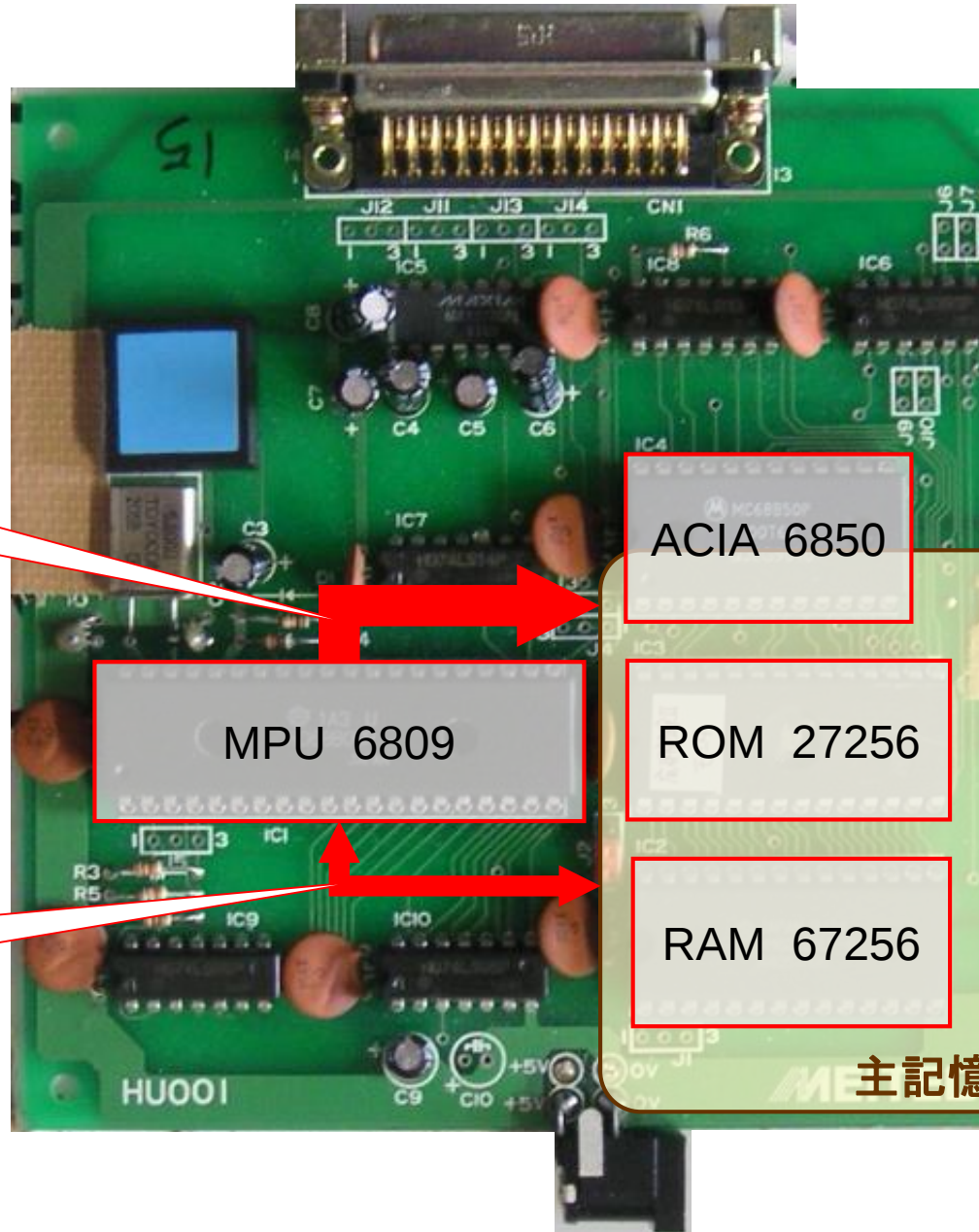
- 単純な構成
- UI装置がない
- 補助記憶装置がない

09brd(まるきゅうボード)の概要 (2/2)



アドレスバス
(16-bit)

データバス
(8-bit)



0000		~	
0010		~	
⋮	⋮		⋮
FFF0		~	

←2-byte→ ←16-byte→
(アドレス)

←1-byte→ ~ ←1-byte→
(データ) (データ)

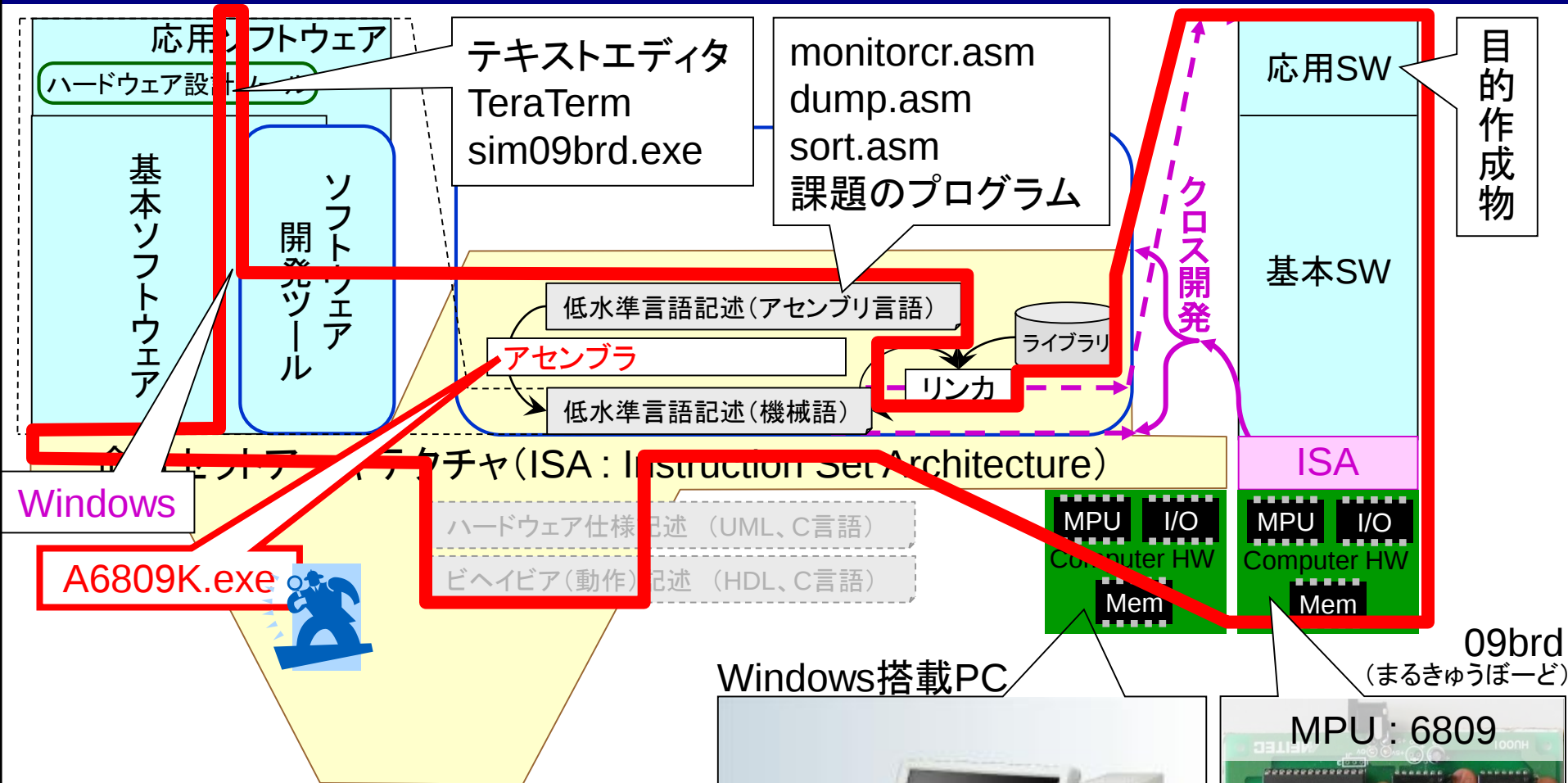
0000	
0001	
⋮	⋮
FFFF	

←16-bit→ ←8-bit→
(アドレス) (データ)

主記憶装置

表現の違い

アセンブラ A6809K.exe



目的
作成物

Windows

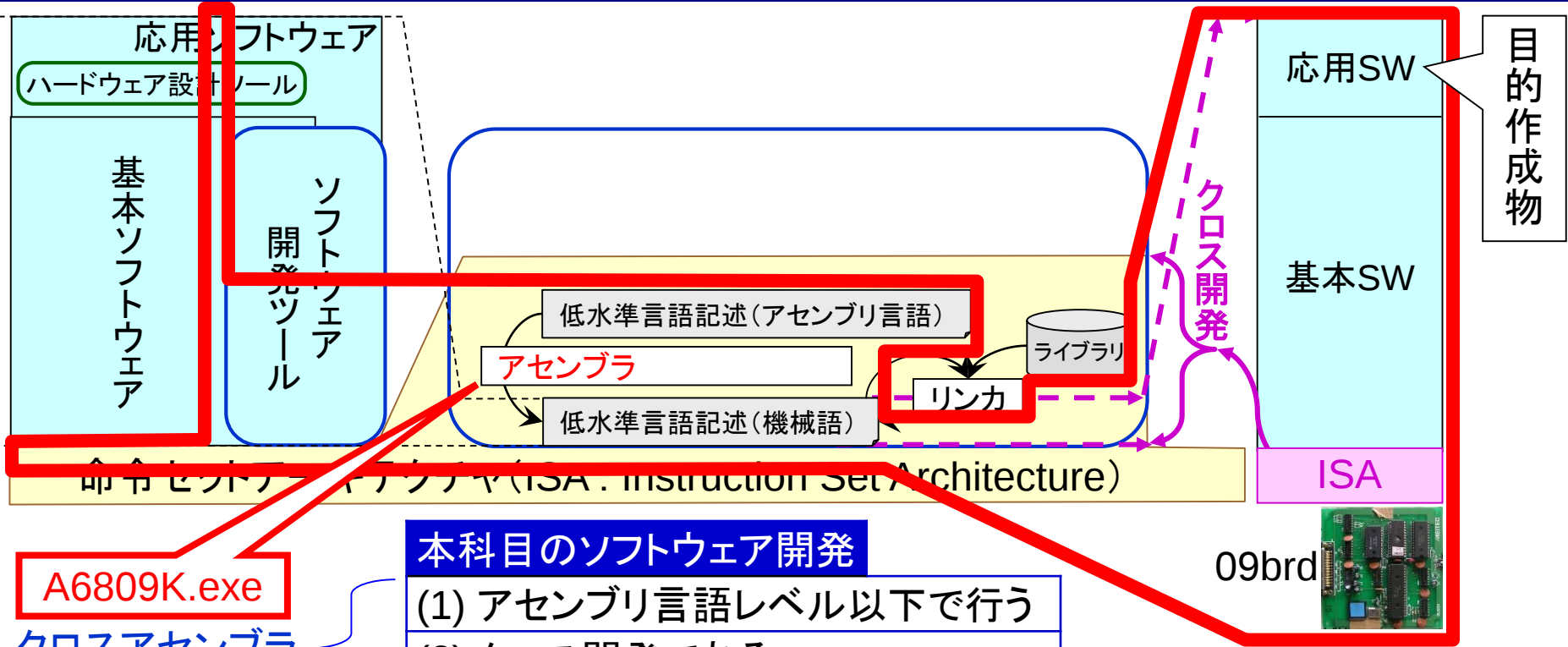
A6809K.exe

Windows搭載PC



【注】
A6809K.exeは、
講義用Webページから取得

ソフトウェア開発環境とA6809K.exe



A6809K.exe

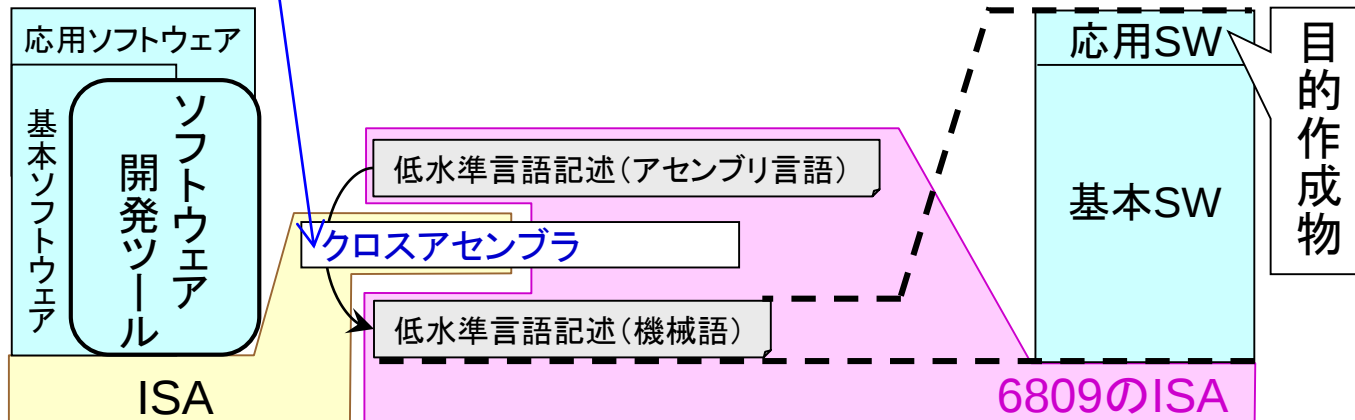
クロスアセンブラ
(クロス開発用アセンブラ)

本科目のソフトウェア開発

- (1) アセンブリ言語レベル以下で行う
- (2) クロス開発である
- (3) リンカを用いない

派生する特徴

- 目的コードのファイル形式が単純
- × モジュール化設計、モジュール再利用開発が出来ない



A6809K.exe が処理する入出力ファイルの種類

.asmファイル: アセンブリ言語プログラムソースファイル

ファイルとしての格納場所

目的作成物

入力側
(1種類)
アセンブラ境界

出力側
(3種類)

応用ソフトウェア
基本ソフトウェア
ソフトウェア
開発ツール

ISA

低水準言語記述(アセンブリ言語)

アセンブラ A6809K.exe

低水準言語記述(機械語)

応用SW

基本SW

6809のISA

補助
記憶
装置

HDD

.lstファイル

.romファイル

.romファイル: 本演習では利用しない

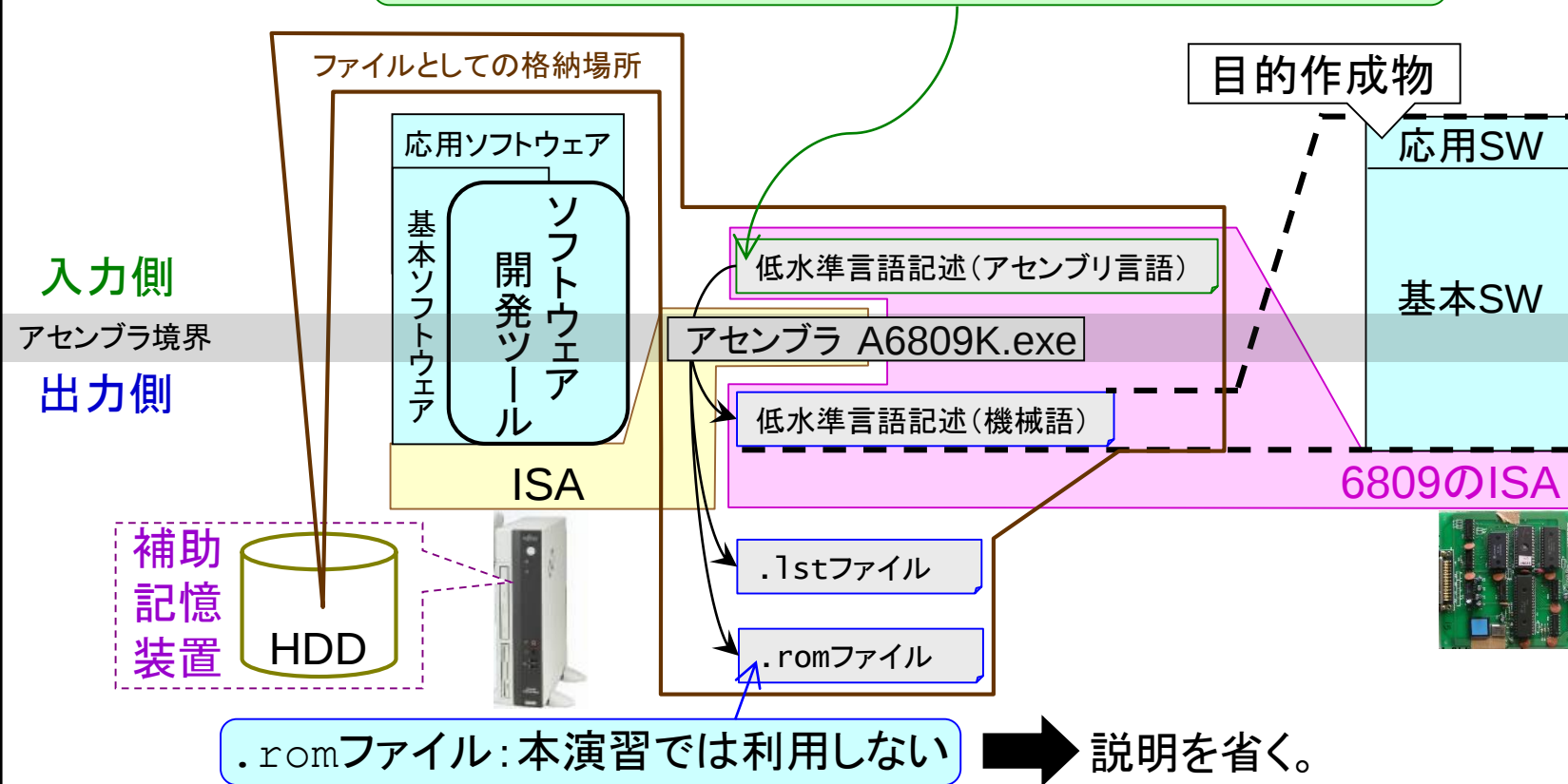
.lstファイル: アセンブル結果のリストファイル(学習のヒント)

.objファイル: 目的作成物。目的コード(実行コード)のファイル

A6809K.exe が処理する入出力ファイルは全4種類であるが、
本演習では .romファイル は利用しないので、実質、3種類のファイルの形式を知ればよい。

A6809K.exeが処理する入出力ファイルの形式(1)

.asmファイル:アセンブリ言語プログラムソースファイル



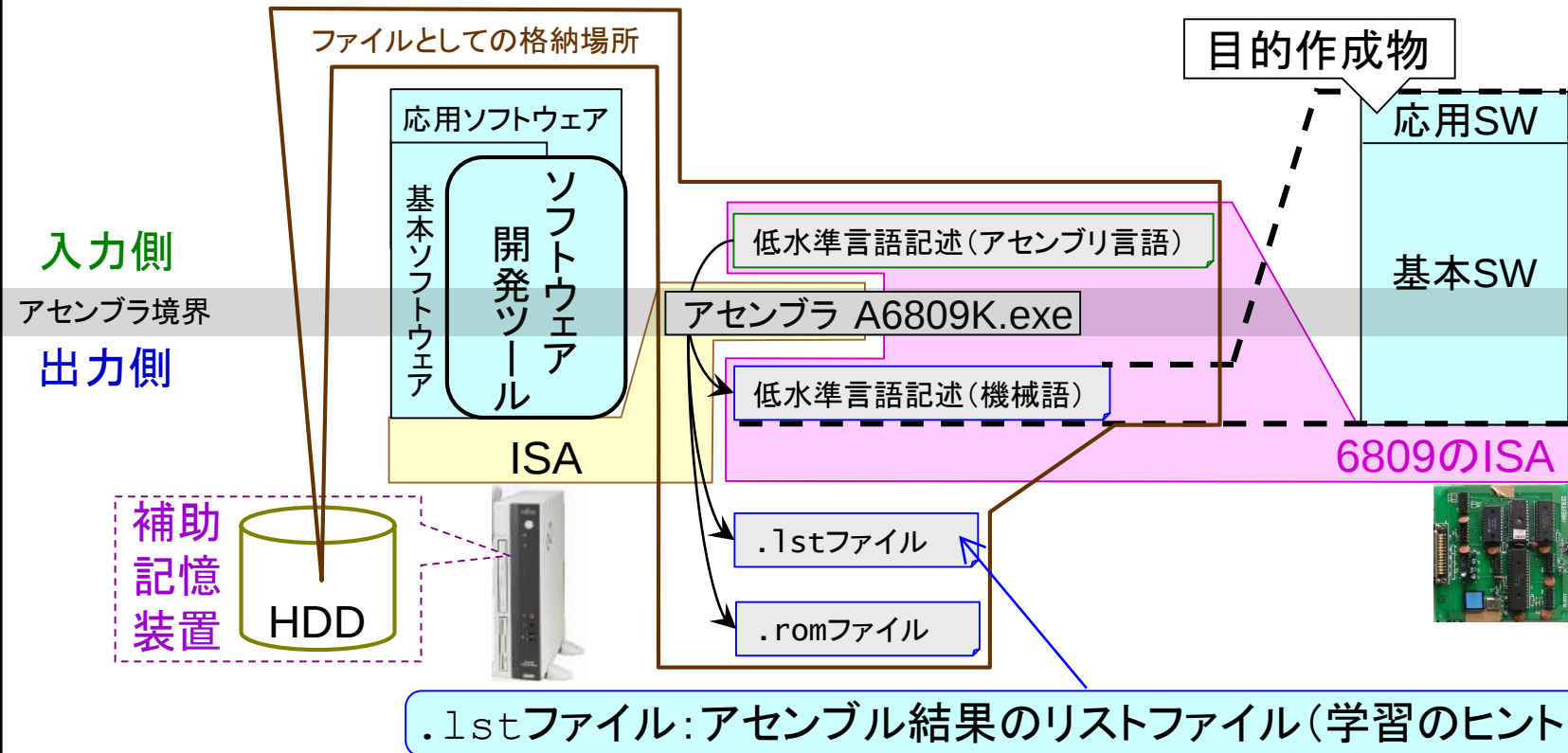
.romファイル:本演習では利用しない ➡ **説明を省く。**

.asmファイルの形式:テキストファイルで、以下が各行の基本形式。詳細はのちほど。

コメント行形式	* 任意文字列
ラベル無・1行形式	◇アセンブリオペコード◇アセンブリオペランド[: ◇アセンブリオペコード◇アセンブリオペランド]
ラベル付・1行形式	ラベル◇アセンブリオペコード◇アセンブリオペランド[: ◇アセンブリオペコード◇アセンブリオペランド]
ラベル付・2行形式	ラベル ◇アセンブリオペコード◇アセンブリオペランド[: ◇アセンブリオペコード◇アセンブリオペランド]

【注】「」部分は省略可、◇は空白1つ以上、その他の用語はのちほど

A6809K.exeが処理する入出力ファイルの形式(2)



.lstファイルの形式: テキストファイルで、以下が各行の形式。

◇通し番号□□(◇数値1)□アドレス□□機械語命令◇(◇数値2)□.asmファイルの1行

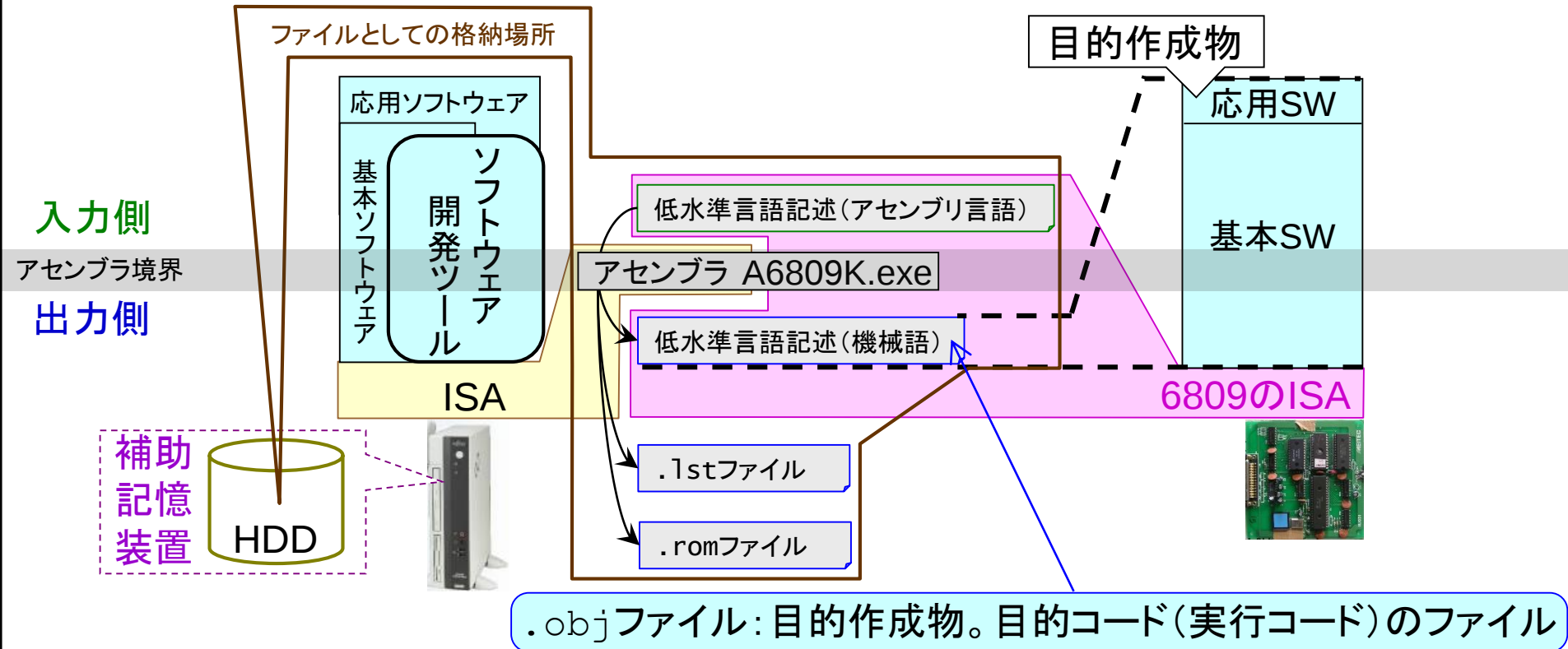
【注】◇は空白1つ以上、□は空白1つ

数値1: 各機械語命令の先頭バイトの、先頭アドレスからの変位バイト数

数値2: 各.asmファイルの1行の、.asmファイル中の行番号

.asmファイルの1行は、改行または:での区切りによる

A6809K.exeが処理する入出力ファイルの形式(3)



.objファイル: 目的作成物。目的コード(実行コード)のファイル

.objファイルの形式: テキストファイル(?)で、以下が全体の形式。

```
@1000[hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh]
@1010[hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh]
      ⋮              ⋮
@1111[hhhhhhhhhhhhhhhhhhhh]—————
```

hhは任意の2桁の16進数。

「」内は機械語命令列。

h2つで1バイト。一行に最大16バイト。

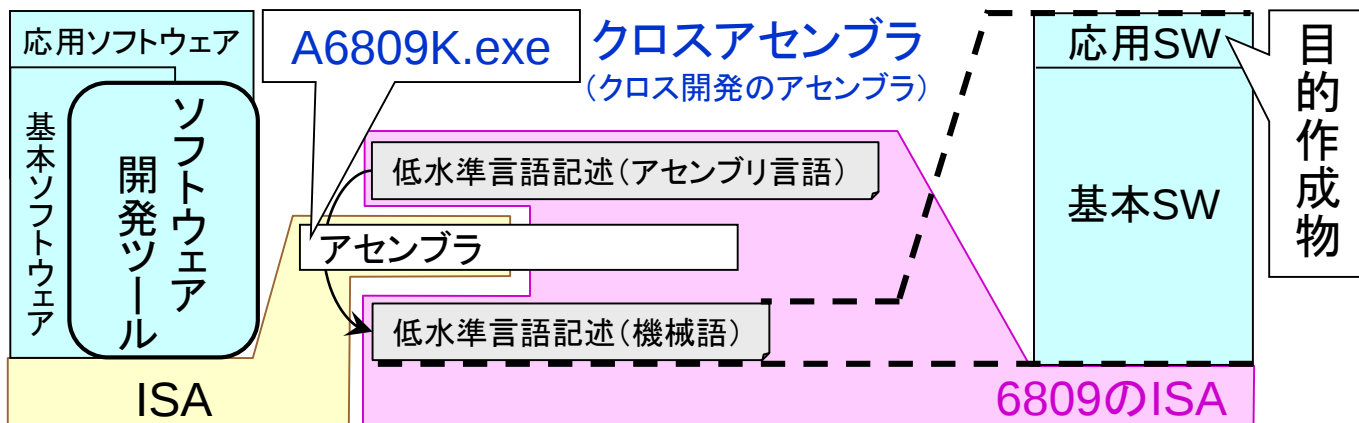
【注】記号「 h 」は以降も使う

E 終端記号

アドレス

末行は、16バイトに満たないことがある。(多くの場合)

クロスアセンブラ A6809K.exe の使い方



【注1】

.asm

: アセンブリ言語
プログラム
ソースファイル。

(1) コマンド
プロンプト
を起動し

Z:\>cd CAE

(2) Change Directory

Z:\CAE>a6809k.exe

(3) クロスアセンブラ A6809K.exe を実行

MC6809 cross assembler start, made by Y. Yoshioka. 2007.8.28

ROM writer format: ASCII/2A (74256 only)

Please input .asm file name = foo.asm

(4) アセンブルしたい

アセンブリ言語プログラム
ファイル名を入力

**** pass 1 start ****

**** pass 2 start ****

** Error count=0 **, end addr=bar

> Generated "foo.lst" file.

> Generated "foo.rom" file.

> Generated "foo.obj" file.

** Assembler completed.

(5) 左のように、

✓ Error count=0

✓ .lst/.rom/.obj の3つのファイルを生成

✓ メッセージ「Assembler completed.」が出力

となれば正常にアセンブル終了。

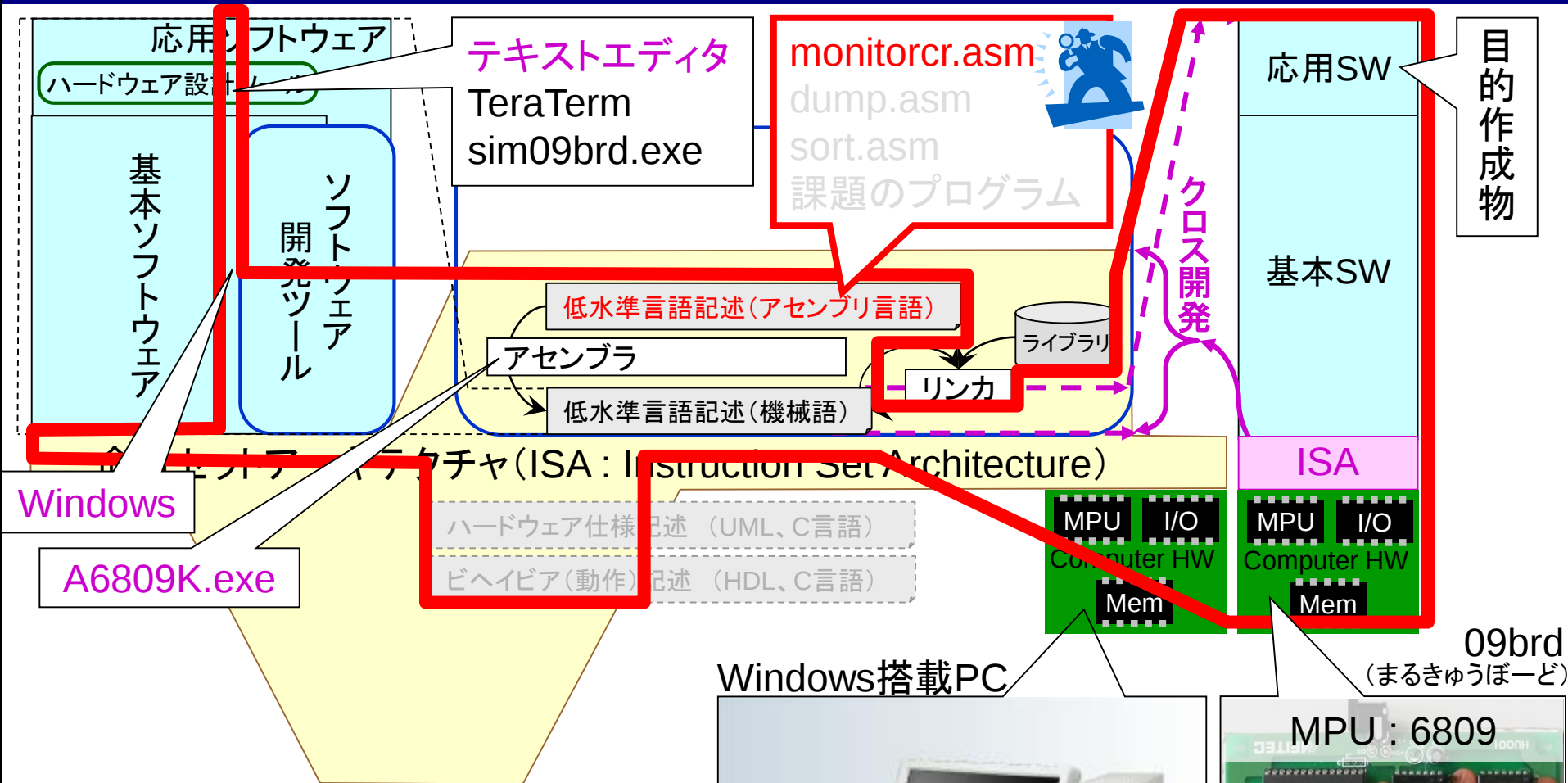
それ以外の場合は、エラー等を修正し、再アセンブルする。

【注2】.obj : 目的作成物。目的コード(実行コード)のファイル。

.lst : アセンブル結果のリストファイル。(学習のヒント)

.rom : 本演習では用いないので無視する。(わざわざファイル削除する必要はない。)

09brd用基本ソフトウェアのソースファイル monitorcr.asm



Windows搭載PC



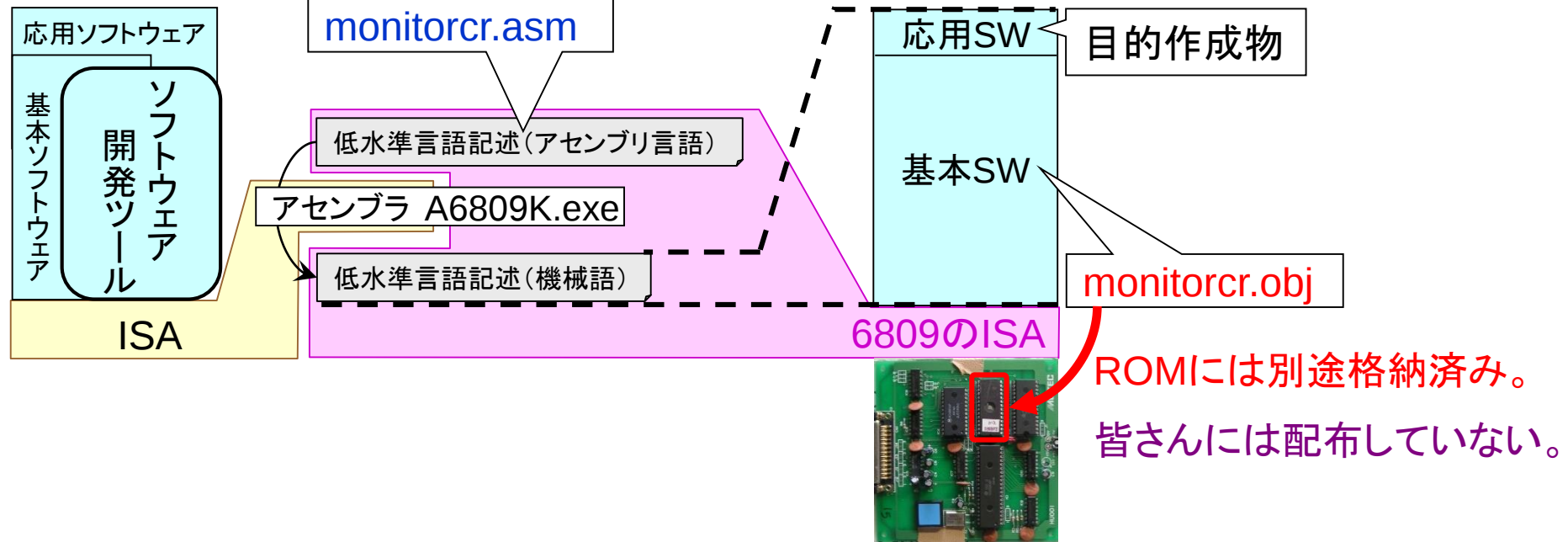
09brd
(まるきゅうぼーど)



【注】
monitorcr.asmは、
講義用Webページから取得

monitorcr.asm と monitorcr.obj

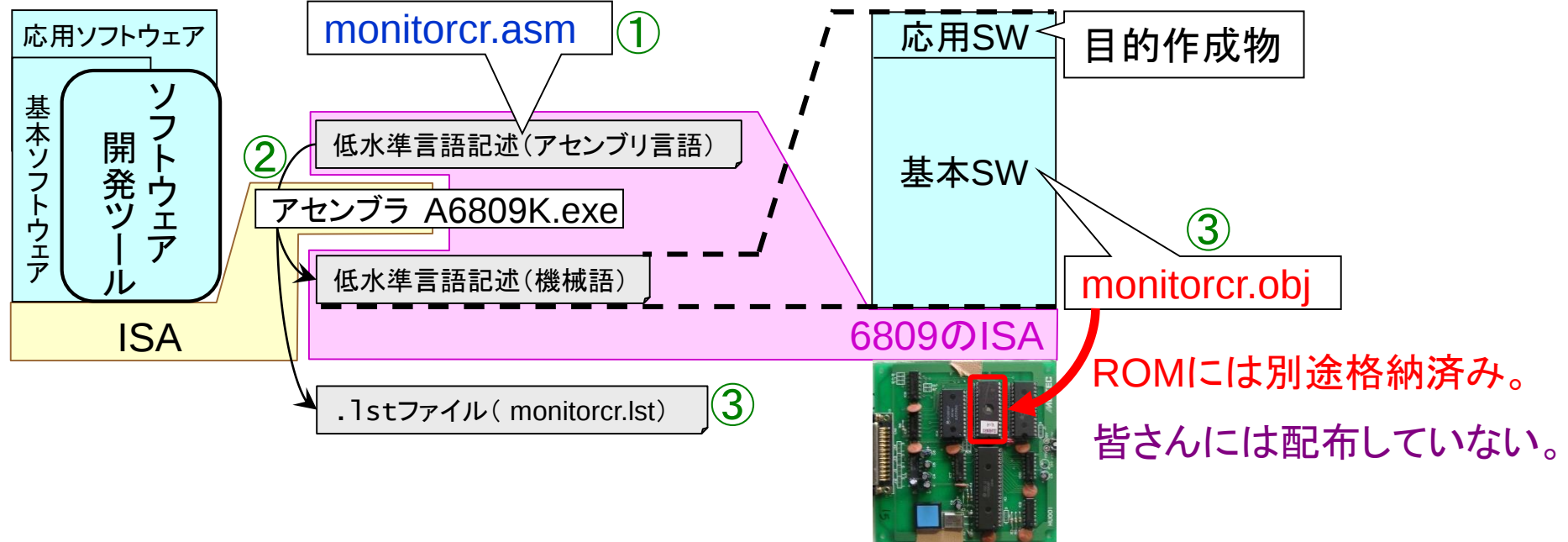
09brdの基本ソフトウェア「(簡易)モニタ」のアセンブリ言語プログラムソースファイル



(簡易)モニタ monitorcr.obj の役割内容	提供コマンド
09brdのメモリ内容をダンプする。	dコマンド
09brdのメモリ内容を編集する。	mコマンド
09brdのメモリに実行コードを転送する。	sコマンド
転送した実行コードを実行する。	gコマンド

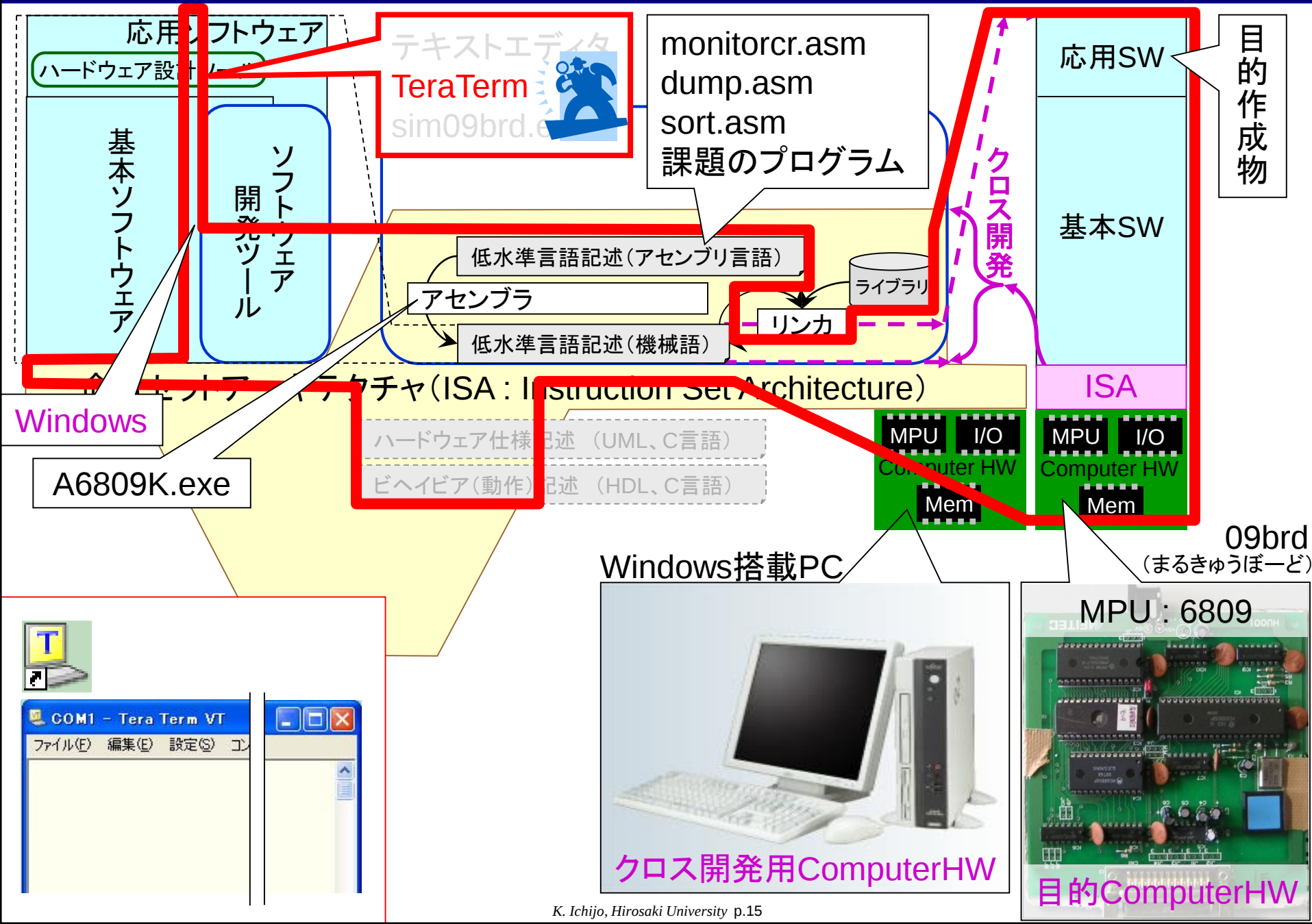
練習

09brdの基本ソフトウェア「(簡易)モニタ」のアセンブリ言語プログラムソースファイル



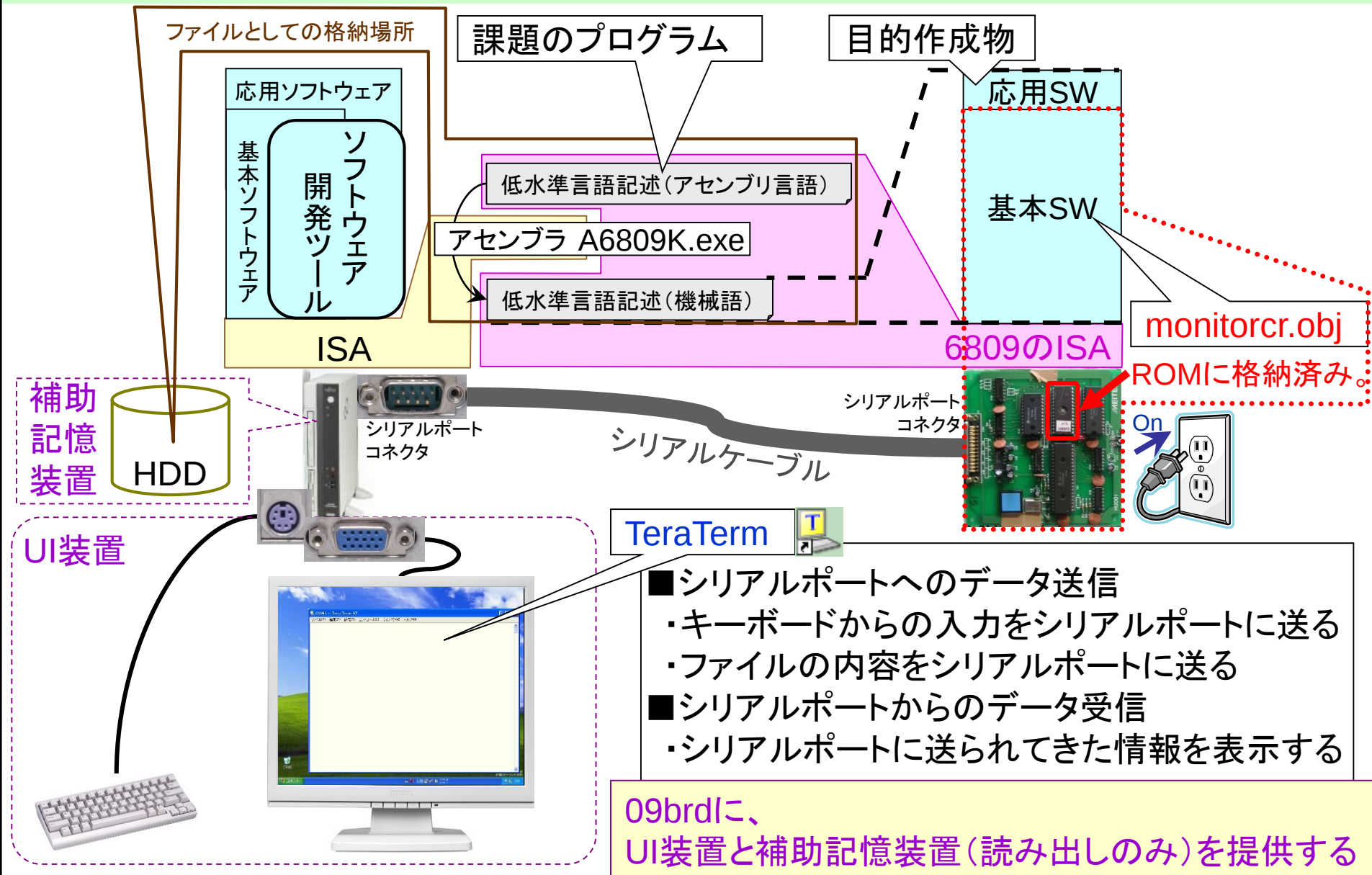
- ① monitorcr.asmをテキストエディタで開いてみる。
- ② monitorcr.asmをアセンブルする。
- ③ 生成された monitorcr.obj、monitorcr.lst をテキストエディタで開いてみる。

TeraTerm(for Windows)



09brdの操作とTeraTermの役割

09brdに、《Q1》モニタコマンドを伝えるには？ 《Q2》目的生成物を送るには？



TeraTermの起動と設定

Tera Term: 新しい接続

○ TCP/IP ホスト(D): myhost.mydomain
[x] ヒストリ(O)
サービス: ○ Telnet TCPポート#(P): 22
○ SSH SSHバージョン(V): SSH2
○ その他 UNSPEC

シリアルのCOM1を選択

☒ シリアル ポート(R): COM1: EX-PC160 Communications

OK キャンセル ヘルプ(H)

COM1 - Tera Term VT

ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)

端末(T)... ウィンドウ(W)... フォント(F)... キーボード(K)... シリアルポート(E)... プロキシ(P)...

Tera Term: 端末の設定

端末サイズ(D): 90 x 35
[x] = ウィンドウサイズ(S):
[] 自動的に調整(W):

改行コード
受信(R): CR
送信(M): CR+LF

端末ID(D): VT100
[x] ローカルエコー(L)

漢字 漢字出力(O): [] B
EUC

ローケル: japanese 言語コード: 932

OK キャンセル ヘルプ(H)

✓ 改行コード>送信: CR+LF
✓ ローカルエコー: On

Tera Term: シリアルポート 設定

ポート(P): COM1
ボー・レート(B): 9600
データ(D): 8 bit
パリティ(A): none
ストップ(S): 2 bit
フロー制御(E): none

OK キャンセル ヘルプ(H)

送信遅延 0 ミリ秒/字(Q) 0 ミリ秒/行(L)

上記の設定に

09brdのリセットボタンを押す

COM1 - Tera Term VT

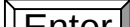
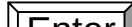
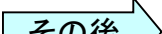
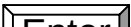
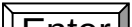
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)

*** Communications monitor *** Made by Y. Yoshioka (2004.01.13)
\$

メッセージと、プロンプト「\$」が表示されれば、設定完了。モニタが起動。

TeraTerm と monitorcr.obj

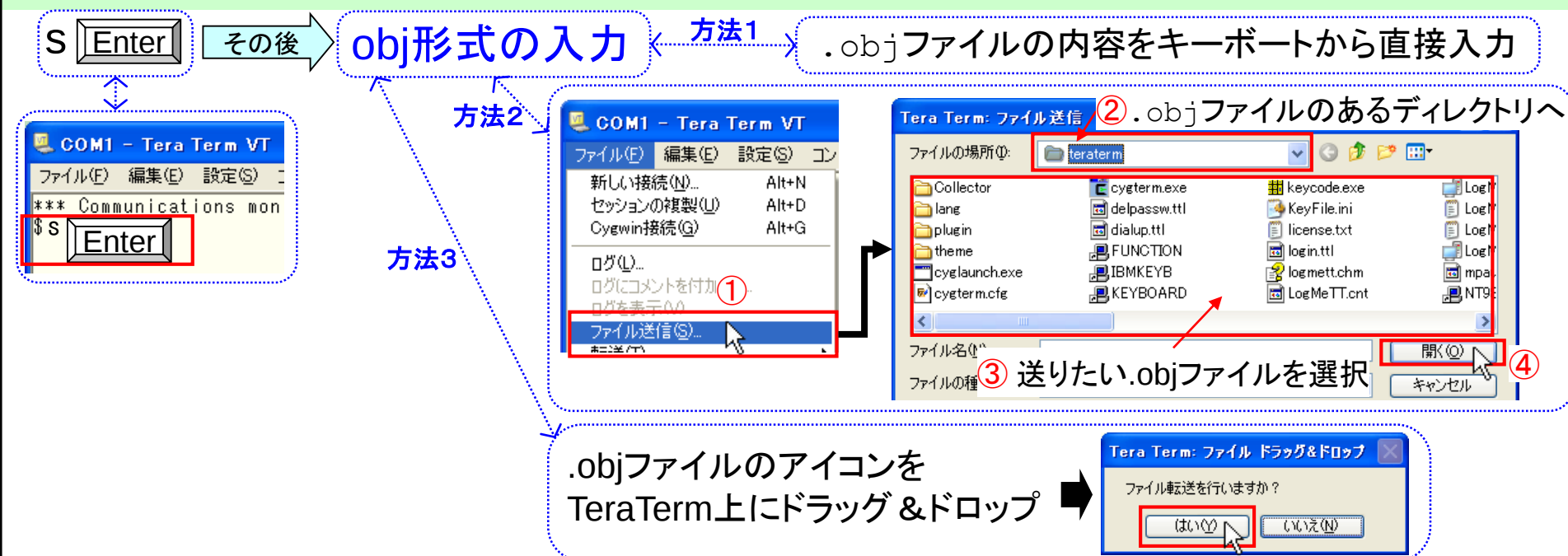
《A1》TeraTerm準備完了→モニタmonitorcr.objのコマンドを09brdへ伝えられる。

コマンドと書式	機能
d[□addr] 	[addrから]256バイトのメモリダンプ
s   obj形式の入力	obj形式データをメモリへ転送
g□addr 	addrからプログラムを実行
m□addr□□hh□hh□...hh 	addrからデータhhを個数分書き込む

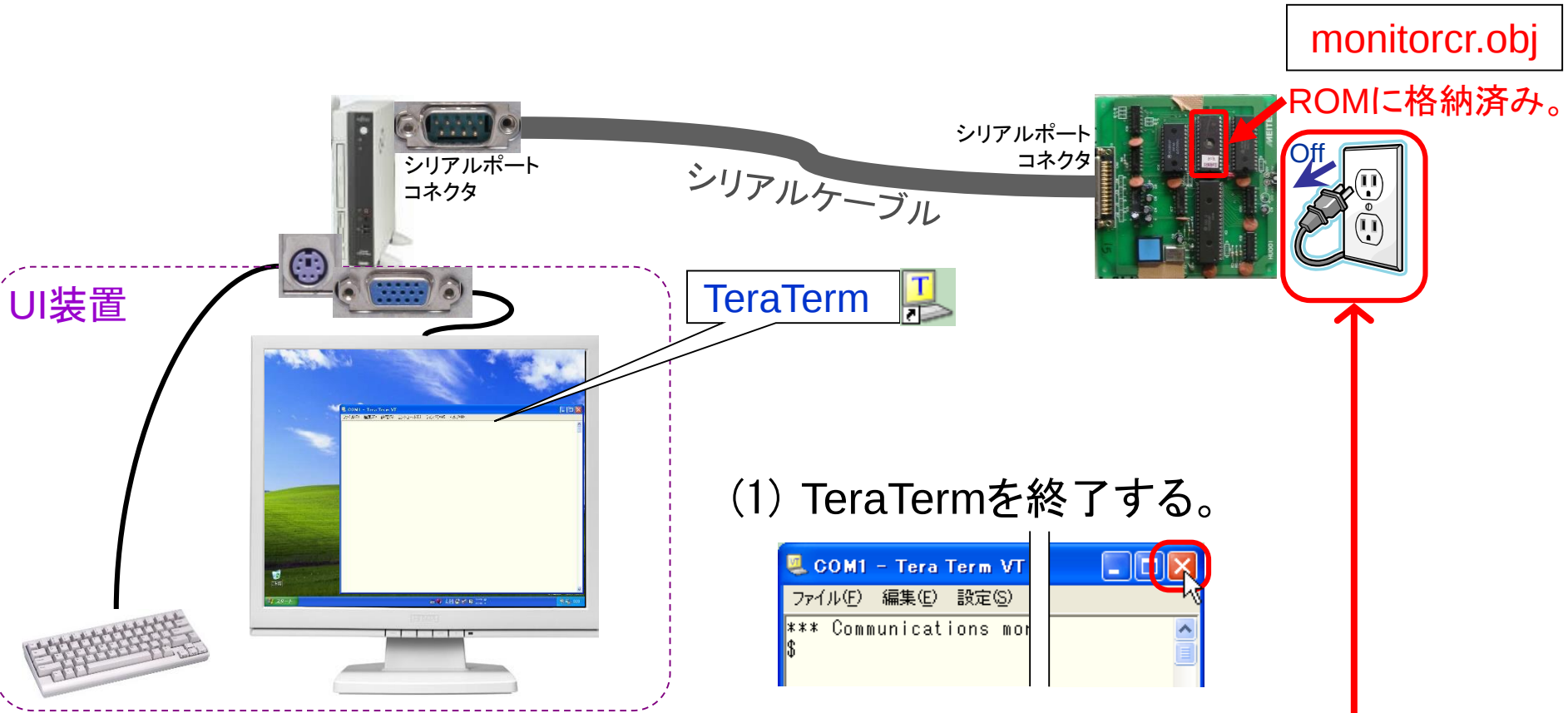
hhは任意の2桁の16進数、個数は最大16個

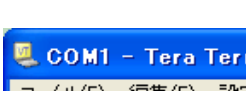
【注】□は空白(スペースキー)1つ
addrは任意のメモリアドレス

《A2.》sコマンドを使用する→目的生成物(.objファイル)を09brdへ送る。



09brd、TeraTermの利用終了



- (1) TeraTermを終了する。
- 
- The screenshot shows the TeraTerm window titled 'COM1 - Tera Term VT'. The menu bar includes 'ファイル(F)', '編集(E)', and '設定(S)'. The main text area displays '*** Communications mor' and '\$'. The window's title bar has standard Windows controls: minimize, maximize, and close. A red circle highlights the close button (X), and a mouse cursor is clicking it.
- (2) 09brdのACアダプタを抜く。
- (3) 09brdと演習室PCとから、シリアルケーブルを外す。

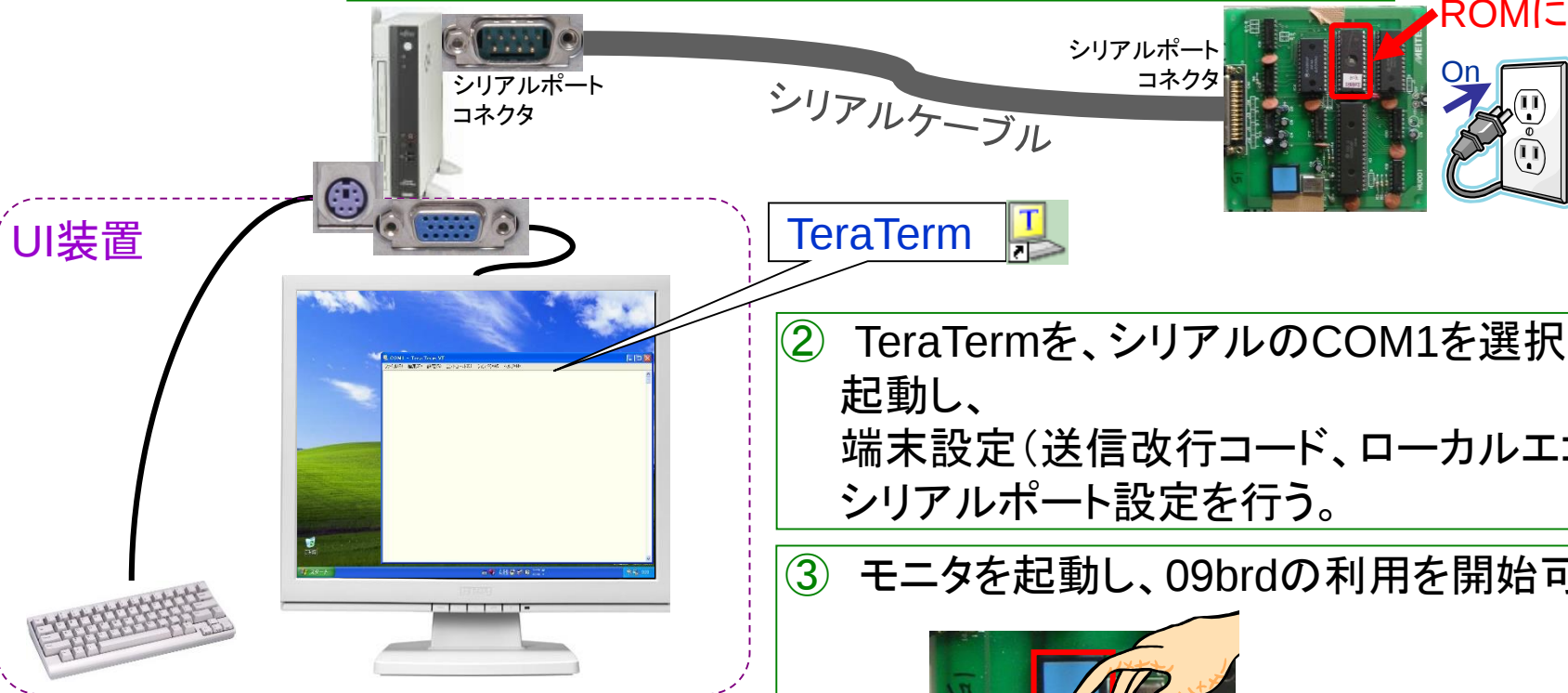
※09brd上のICの発熱に注意すること。

練習

- ① 09brdと演習室PCとを、シリアルケーブルで接続し、09brdのACアダプタを差し込む。

monitorcr.obj

ROMに格納済み。



- ② TeraTermを、シリアルのCOM1を選択して起動し、端末設定(送信改行コード、ローカルエコー)とシリアルポート設定を行う。

- ③ モニタを起動し、09brdの利用を開始可とする。



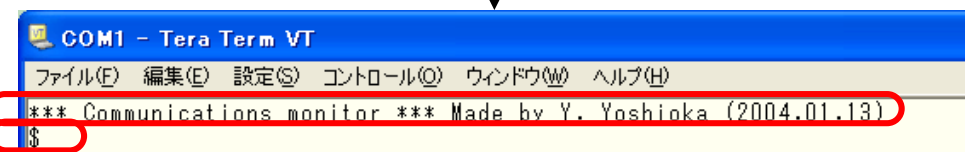
09brdのリセットボタンを押す

- ④ dコマンドを実行してみる。

d

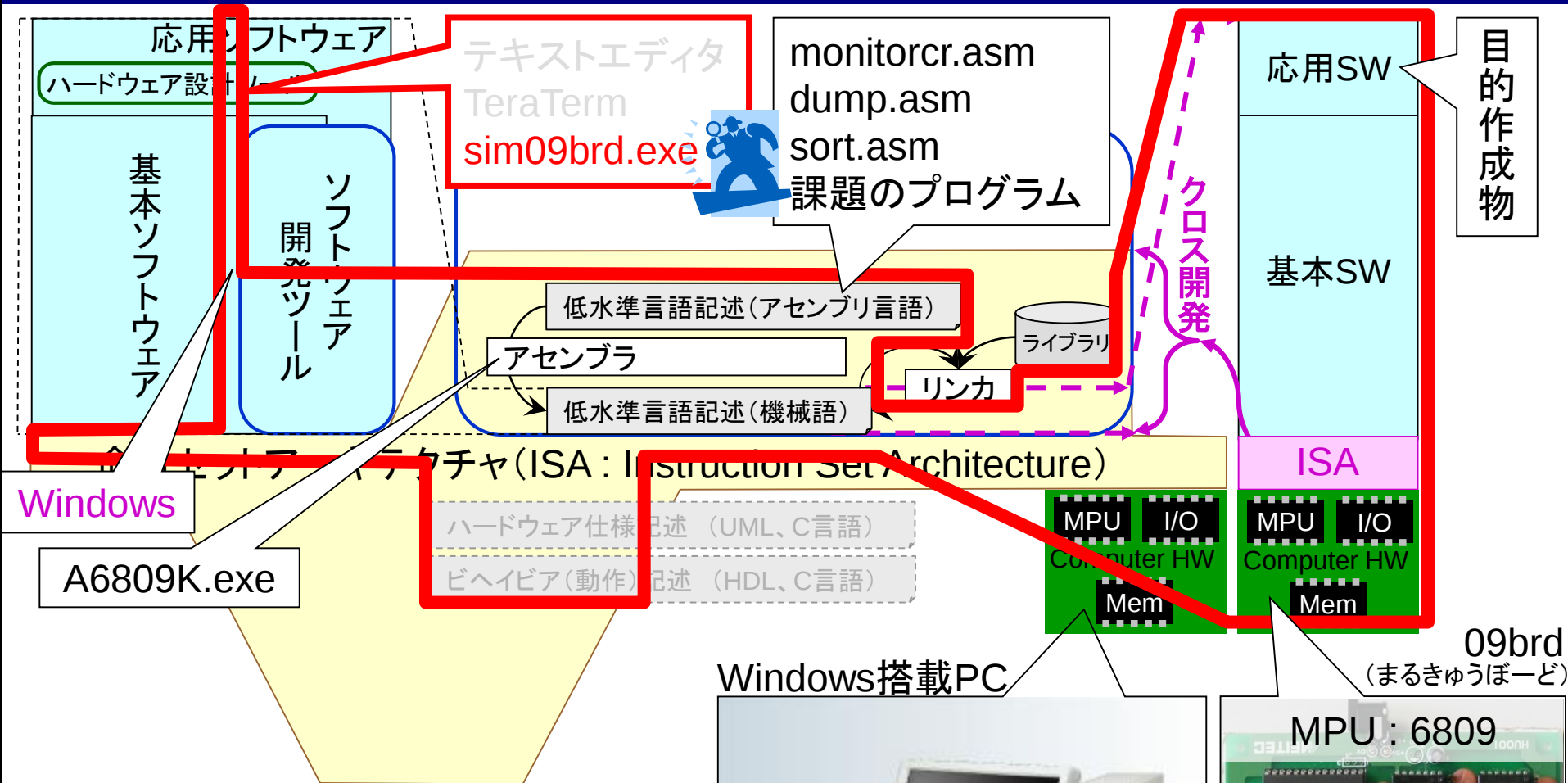
d□1000

など...



メッセージと、プロンプト「\$」が表示されれば、設定完了。モニタが起動。

sim09brd.exe



【注】
sim09brd.exeは、
講義用Webページから取得

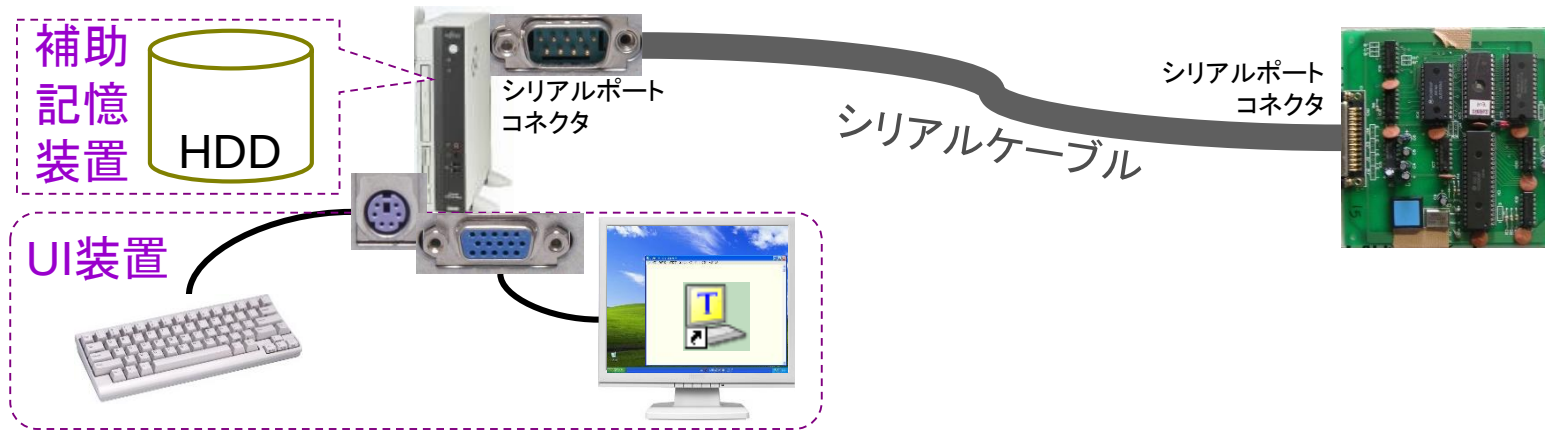
Windows搭載PC



09brd
(まるきゅうぼーど)



sim09brd.exe の役割と起動



sim09brd.exe

上図のような環境を
演習室PC上でシミュレートするソフトウェア
→全員使える。

- (1) **monitorcr.obj** を sim09brd.exe と同じディレクトリに用意しておく。
- (2a) コマンドプロンプトを起動→Change Directory→sim09brd.exeを実行

```
Z:\>cd CAE 
```

```
Z:\>CAE>sim09brd.exe 
```

```
*** Communications monitor *** Made by Y. Yoshioka (2004.01.13)  
$
```

- (2b) sim09brd.exe をダブルクリック

```
*** Communications monitor *** Made by Y. Yoshioka (2004.01.13)  
$
```

・メッセージ
・プロンプト「\$」
が表示され、モニタ起動。

sim09brd.exe の操作

09brdのモニタコマンド類似のもの

コマンドと書式	機能
d[addr] Enter	[addrから]256バイトのメモリダンプ
s Enter その後 obj形式の入力	obj形式データをメモリへ転送
g addr Enter	addrからプログラムを実行
m addr hh hh ... hh Enter	addrからデータhhを個数分書き込む

hhは任意の2桁の16進数、個数は最大16個

【注】は空白(スペースキー)1つ
addrは任意のメモリアドレス

※sコマンドについて、**sim09brdではTeraTermと同様にはできない。**
∴sim09brd≠TeraTerm ただし、キーボードからの直接入力是可以る。

sim09brdのみに装備されているモニタコマンド

コマンドと書式	機能
? Enter	ヘルプ
Q Enter	シミュレータ終了
R Enter	リセットボタンの替わり
C Enter	0x1000～0x7FFFのメモリ内容を消去
S foo.obj Enter	foo.obj ファイルの内容をメモリへ転送

代わりに

練習

① monitorcr.obj と sim09brd.exe とを、同じディレクトリ「CAE」に置く。

② 次のいずれかの方法で、sim09brd.exeを実行する。

(a) コマンドプロンプトを起動後、以下のようにコマンド入力

```
Z:¥>cd CAE   
Z:¥CAE>sim09brd.exe   
*** Communications monitor *** Made by Y. Yoshioka (2004.01.13)  
$
```

(b) sim09brd.exe をダブルクリック

```
*** Communications monitor *** Made by Y. Yoshioka (2004.01.13)  
$
```

③ dコマンドを実行してみる。

```
d   
d 1000  など... 【注】は空白(スペースキー)1つ
```

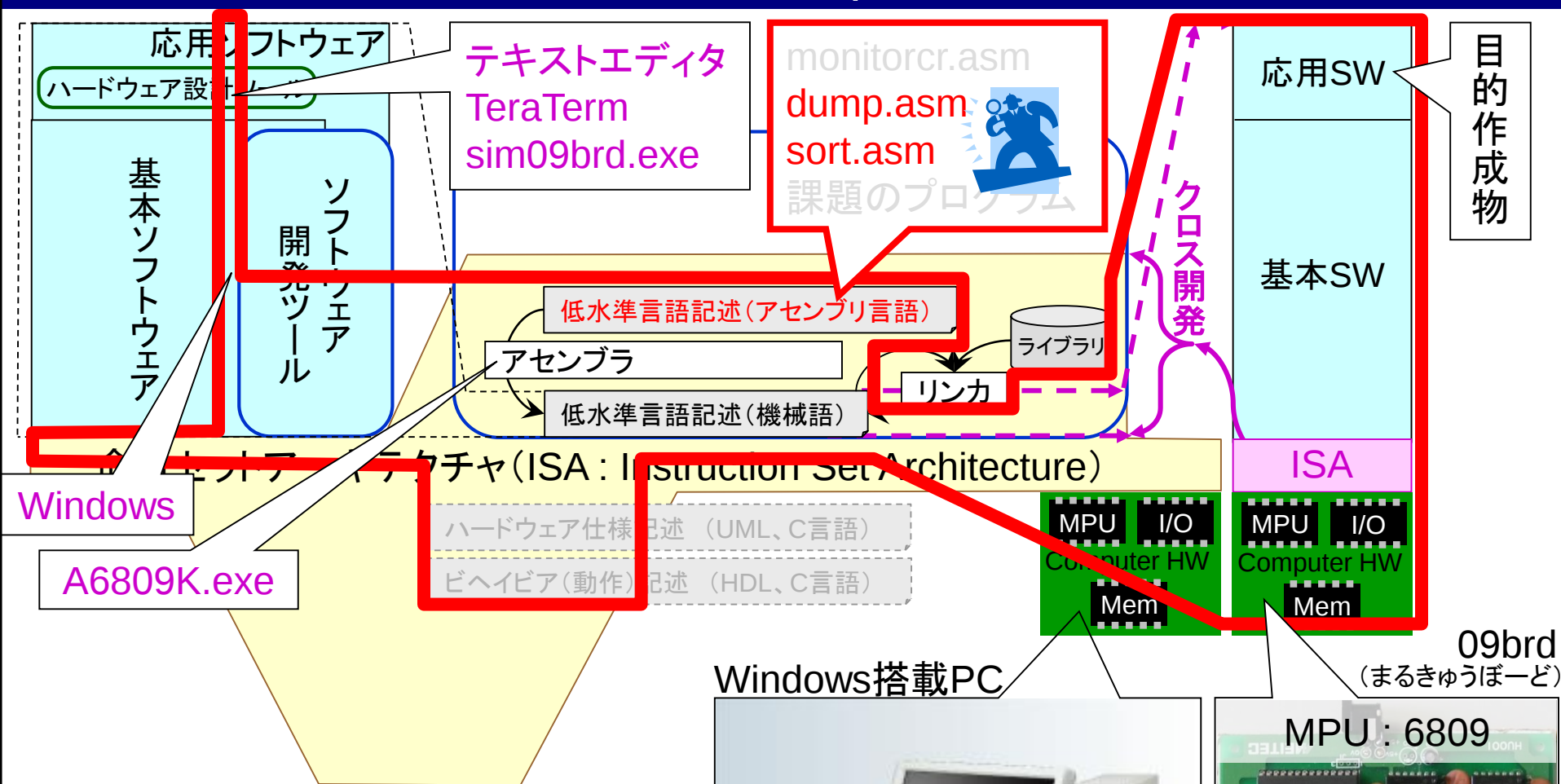
④ ?、R、Cコマンドを実行してみる。

⑤ Qコマンドでsim09brd.exeを終了する。

以前の練習で扱った、モニタプログラムについて、

- ① monitorcr.lst、monitorcr.obj をテキストエディタで開き、
monitorcr.objはメモリのどこからどこに転送されるのか確認する。
(転送先の先頭をaddrB、末尾をaddrE、とする。)
- ② 09brdまたはsim09brdで、addrB以降の256バイトを見て、
テキストエディタ上のmonitorcr.objの内容と比べてみる。
- ③ 09brdまたはsim09brdで、addrE以前の256バイトを見て、
テキストエディタ上のmonitorcr.objの内容と比べてみる。

サンプルプログラム dump.asm と sort.asm



【注】
dump.asmとsort.asmは、
講義用Webページから取得

Windows搭載PC



練習

2つのサンプルプログラム、dump.asm(メモリダンププログラム)、
sort.asm(データソートプログラム)、

それぞれについて、

- ① .asm ファイルをテキストエディタで開く。
- ② アセンブルする。
- ③ 生成された、.lst、.obj をテキストエディタで開く。
- ④ .obj はメモリのどこからどこに転送されるのか確認する。
(転送先の先頭をaddrB、末尾をaddrE、とする。)
- ⑤ 09brd または sim09brd で .obj を以下の手順で実行してみる。
 - (a) .obj 転送前の addrB～addrE の内容を見る。
 - (b) .obj を転送する。
 - (c) .obj 転送後(実行前)の addrB～addrE の内容を見る。
 - (d) .obj を実行する。
 - (e) .obj 実行後の addrB～addrE の内容を見る。
- ⑥ .asm または .lst を読んで理解し、実行結果を考察する。

練習

サンプルプログラムsort.objについて、
前の練習で調べた、sort.objの転送先の先頭および末尾のアドレスを
それぞれ、addrB、addrEと置く。

09brdまたはsim09brdで、sort.obj を以下の手順で実行してみる。

- (a) .obj 転送前の addrB～addrE の内容を見る。
- (b) .obj を転送する。
- (c) .obj 転送後(実行前)の addrB～addrE の内容を見る。
- (d) .obj を実行する。
- (e) .obj 実行後の addrB～addrE の内容を見る。
- (f) mコマンドを実行する。

【例】

m□1070□□34□a3□e0□08□21□14□89□5c□27□62□25□42□20□41□9d

ただし、□は空白(スペースキー)1つ

- (g) .obj 実行前の addrB～addrE の内容を見る。
- (h) .obj を実行する。
- (i) .obj 実行後の addrB～addrE の内容を見る。