

タイトル

初歩の組み込みプログラム

わんくま名古屋勉強会

• 自己紹介

– 名前

- あんどちん(安藤敏彦)

– 所在地

- 千葉県

– 職業

- プログラマ(主に組込系)

– 覚えたい言語

- 英語

組み込みプログラムとは

- 機器の制御を行うためのプログラム
 - ターゲットの種類は様々
 - CPUは4bit～64bit等様々
 - メモリのサイズも数KB～100MB超まで様々
 - 主に使われるOS
 - 無し
 - ITRON (T-Kernel)
 - Linux
 - Windows CE
 - 開発言語は主にC/アセンブラ(C++/Javaも)

組み込みの魅力

- 光る・回る・音がする
 - モーターなど実際に目の前で物が動く
 - LEDなどが実際に目の前で光る
 - PCM再生など実際に音が出る

組み込みやっけて良かったこと

- ハードのことが分かるようになる
 - 躊躇なくPCの改造ができる
- その結果
 - ZaurusのMD->CF
 - EeePCのSSD->CF/大容量SSD

組込みが難しいと思われる要素

- 制約が大きい

- 未だにRAM/ROMが数KBなどという開発もある。
 - 場合によってはアセンブラが必須となる
- タイミングがシビア
 - μ 秒単位の制御を行う必要がある場合もある



開発方法

- 開発言語

- アセンブラ

- 小規模開発ではフルアセンブラもある
 - 大規模開発でもブートストラップ部分はアセンブラ

- C/C++

- 主流開発言語
 - 大規模開発ではデバイスドライバ・ミドルウェアなど

- Java

- 携帯電話のアプリケーションなど

開発環境

- 開発対象により開発環境は色々
 - エディタ+コンパイラ+デバッガ(ICE)
 - Eclipse
 - Visual Studio (eMbedded VC)
 - Hew (Renesas)
 - MULTI (ADAC)
 - eBinder (eSOL)

今回のセッションで使用する環境

- AKI H8-3052 + AKI-USB
 - 秋月電子通商から販売されているH8/3052CPUを実装した基板とUSBボード
 - CPU
 - H8/3052 (25MHz)
 - RAM
 - 8KB (+128K)
 - ROM
 - 512KB

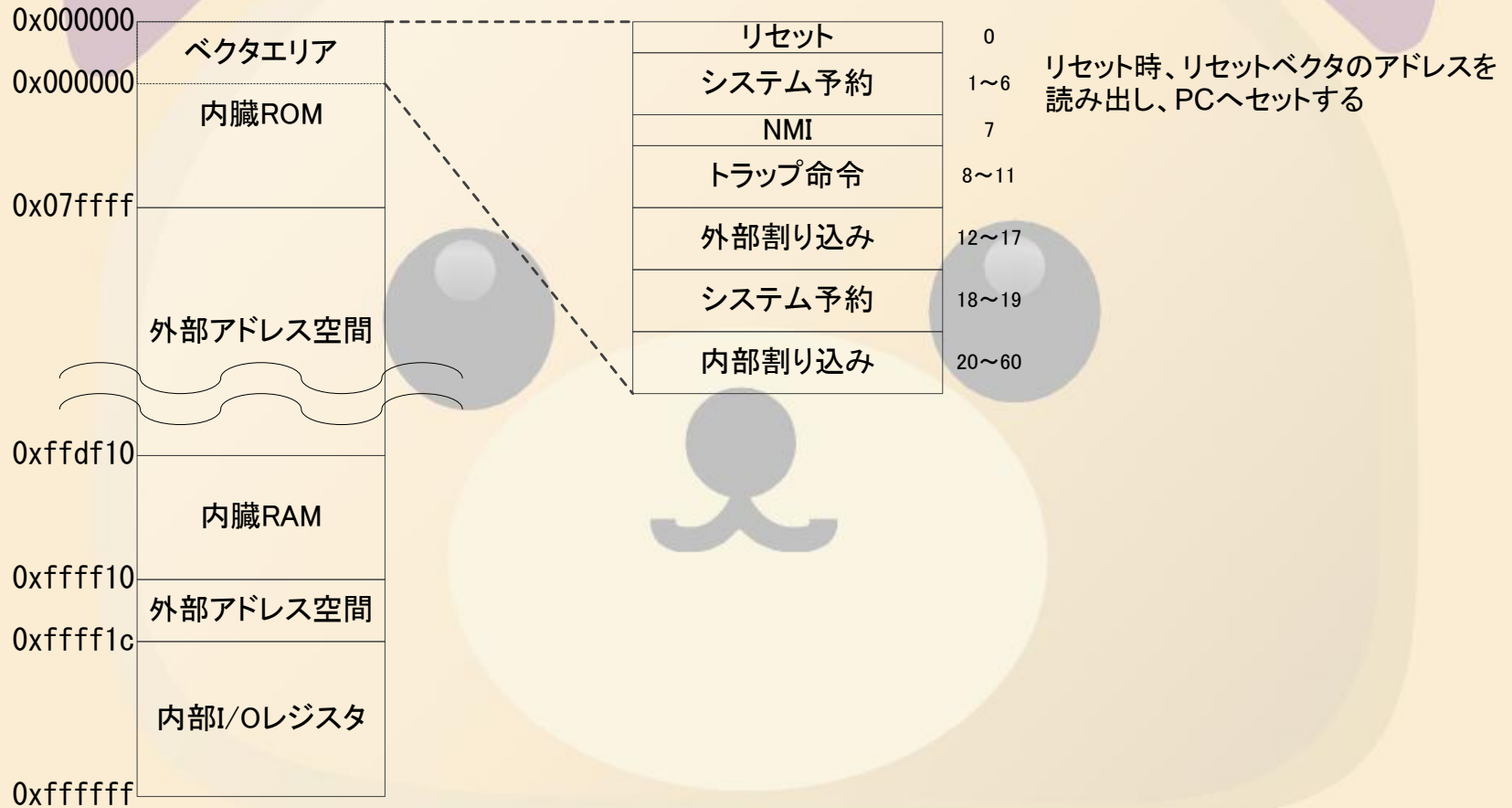
H8/3052のCPUコア・H8/300Hの概要

- 概要

- 16bit × 16本のレジスタセット
 - 8bit × 16/32bit × 8としても使用可
- 62種類の命令セット
- 8種類のアドレッシングモード

H8/3052のメモリマップ

H8/3052 モード6のメモリマップ



エンディアンの違い

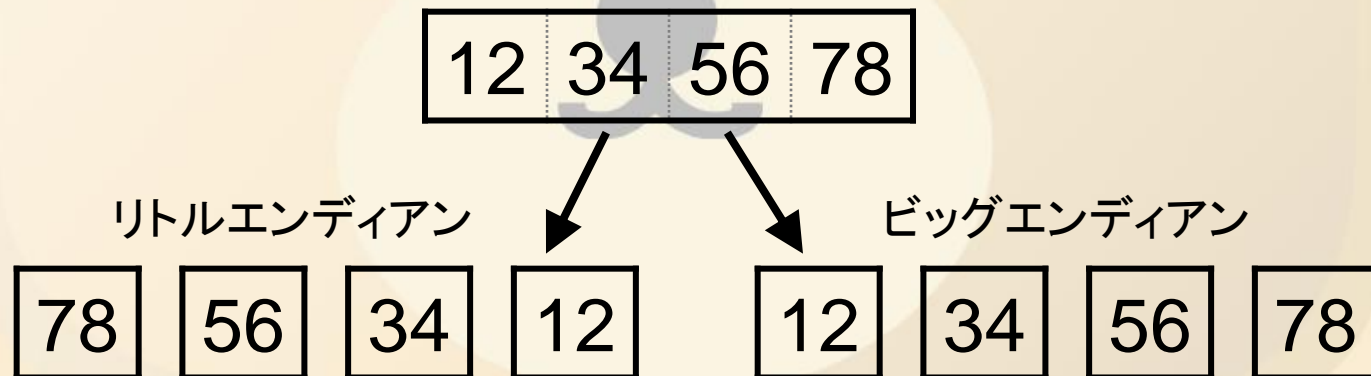
- エンディアン

- リトルエンディアン

- 最下位バイトから最小のアドレスに格納される

- ビッグエンディアン

- 最上位バイトから最小のアドレスに格納される



H8/3052のCPUコア・H8/300Hの概要

レジスタ構成

汎用レジスタ

ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7(SP)	E7	R7H	R7L

コントロールレジスタ

PC

CCR

I	UI	H	U	N	Z	V	C
---	----	---	---	---	---	---	---

命令セット

機能	命令								
データ転送	MOV	POP	PUSH	MOVFPPE	MOVTPPE				
算術演算	ADD	CMP	SUB	ADDX	SUBX	ADDS	SUBS	INC	DEC
	MULXU	MULXS	DIVXU	DIVXS	NEG	EXTU	EXTS		
論理演算	AND	OR	XOR	NOT					
シフト	SHAL	SHAR	SHLL	SHLR	ROTXL	ROTXR	ROTL	ROTR	
ビット操作	BSET	BCLR	BNOT	BTST	BLD	BILD	BST	BIST	BAND
	BIAND	BOR	BIOR	BXOR	BIXOR				
分岐	Bcc	BSR	JMP	JSR	RTS				
システム制御	TRAPA	RTE	SLEEP	LDC	STC	ANDC	ORC	XORC	NOP
ブロック転送	EEPMOV								

• アセンブラでの表記

– 命令 ソース, デスティネーション

- 例) MOV R0, R1



アドレッシングモード

記号	アドレッシングモード
Rn	レジスタ直接
@ERn	レジスタ間接
@(d:16,ERn)/@(d:24,ERn)	ディスプレイースメント(16/24ビット)付レジスタ間接
@ERn+/@-ERn	ポストインクリメント/プリデクリメントレジスタ間接
@aa:8/@aa:16/@aa:24	絶対アドレス(8/16/24ビット)
#xx:8/#xx:16/#xx:32	イミディエイト(8/16/32ビット)
@(d:8,PC)/@(d:16,PC)	プログラムカウンタ相対(8/16ビット)
@@aa:8	メモリ間接

インストラクションマップ・1ワード目上位バイト

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	EX01	STC	LDC	ORC	XORC	ANDC	LDC	ADD		EX0A	EX0B	MOV		ADDX	EX0F
1	EX10	EX11	EX12	EX13	OR	XOR	AND	EX17	SUB		EX1A	EX1B	CMP		SUBX	EX1F
2	MOVB															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MLXU	DVXU	MLXU	DVXU	RTS	BSR	RTE	TRPA	EX58	JMP		BSR		JSR		
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV							
7	BSET	BNOT	BCLR	BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	EX79	EX7A	EPPM	X7C0	X7D0	X7EX	X7FX
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

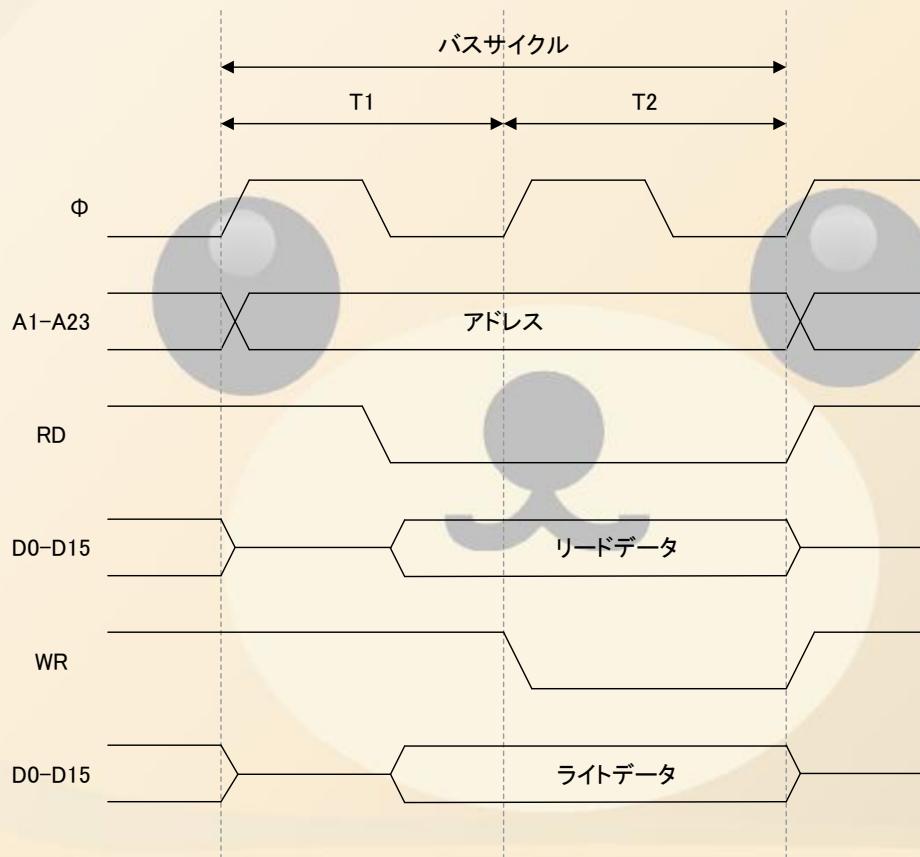


インストラクション長

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	*	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	2	1	2	1	2	1	2	1
6	1	1	1	1	1	1	1	1	1	1	*	*	1	1	2	2
7	1	1	1	1	1	1	1	1	4	2	3	2	2	2	2	2
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

バスタイミング

内部メモリアクセスの場合



開発環境の構築

- 開発環境(gcc)の作成

- binutils/gccをダウンロード/展開

- (<http://www.gnu.org/software/binutils/> <http://gcc.gnu.org/>)

- binutilsの作成(binutils2.16迄)

- `configure --target=h8300-hms --prefix=[installed directory]`

- gccの作成(gcc3.x迄)

- `configure --target=h8300-hms --prefix=[installed directory] --with-gnu-as --with-gnu-ld --enable-languages=c`

ポート設定

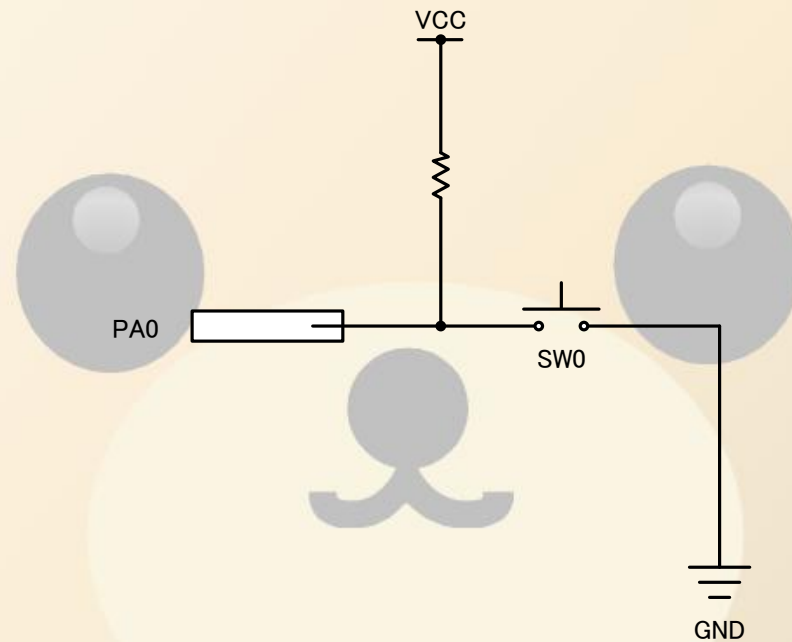
PA	方向	アサイン
7	NC	
6	NC	
5	NC	
4	NC	
3	IN	SW3
2	IN	SW2
1	IN	SW1
0	IN	SW0

PB	方向	アサイン
7	OUT	LCD-E
6	NC	
5	NC	
4	OUT	LCD-RS
3	OUT	LCD-DB7/LED3
2	OUT	LCD-DB6/LED2
1	OUT	LCD-DB5/LED1
0	OUT	LCD-DB4/LED0



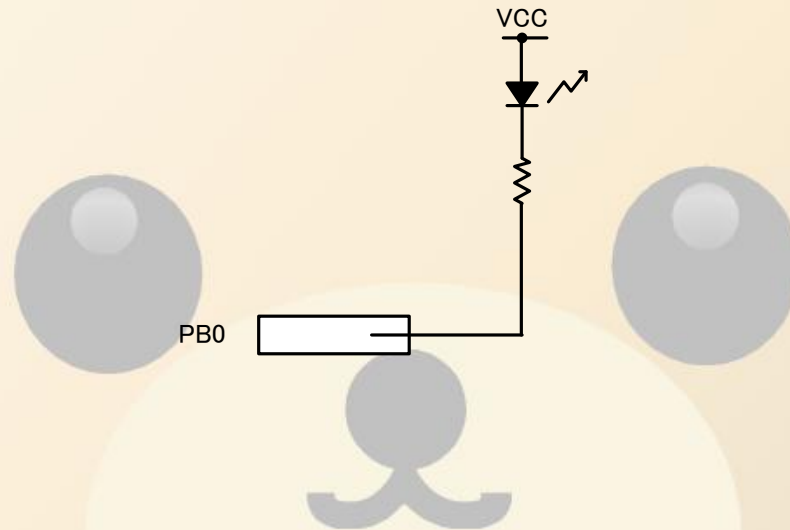
入力ポート

- スイッチはPA0～PA3に接続されている



出力ポート

- LEDがPB0～PB3に接続されている



ポートの初期化

- IOポートのレジスタ構成

レジスタ名称	R/W	機能
データディレクションレジスタ	W	ポートの入出力方向設定
データレジスタ	R/W	ポート入出力

スタートアップの作成

- リセットベクタを設定する
- .data/.bssを初期化
- mainを呼び出す

Main以降の処理

- ポートA0～A3を入力にする
- ポートB0～B3を出力にする
- ポートAから入力
- ポートBへ出力

ハードのみで実現すると

- ソフトで制御するより圧倒的に簡単で低コスト

