Intel 32bit アーキテクチャー

コンピュータ基礎 (8)

菊池浩明

講義概要

- 教科書
  - 4章プロセッサ
  - 5. 動作の流れ
  - 6. 実際のプロセッサー
    - 8086アーキテクチャー
    - 80386(IA32)アーキテクチャー

1. Intelアーキテクチャー

- 8080
  - 8 bit, 1974
- _____
  - 16bit, 1978, x86
- 80386
  - 32bit, 1985, IA-32
- Core-2
  - 64bit, 2006, x64

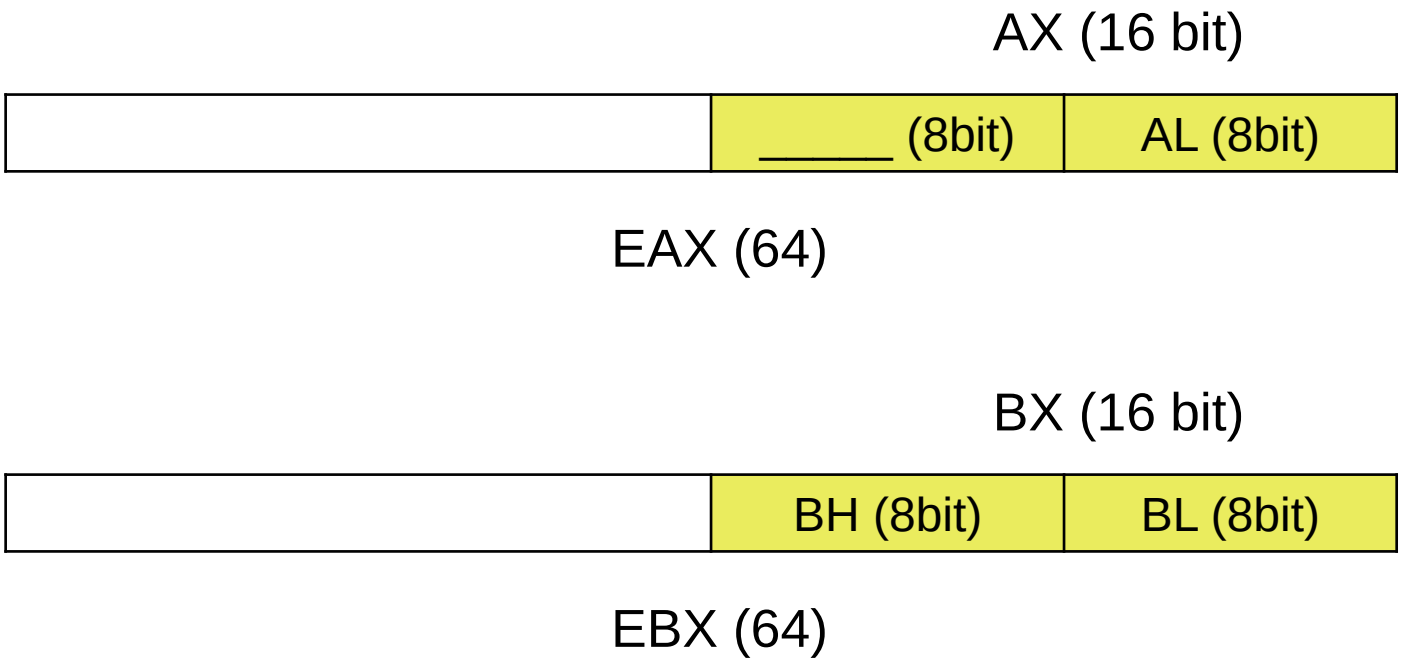
プロセッサー構成要素

- ALU
- Register
- Bus

レジスター register

- 8086 (16bit)
  - AX Accumulator reg.
  - BX Base address
  - CX Count
  - DX Data
  - SI Source
  - DI Destination
- 80386 (IA32)
  - EAX (Extended)
  - EBX
  - ECX
  - EDX
  - ESI
  - EDI
  - ESP (Stack)
  - EBP (Stack Flame)

32-bit 16-bitレジスタの関係



フラグ

- 主なFlag register
  - OF Overflow
  - DF Direction
  - SF Sign Flag
  - ZF Zero Flag
  - PF Parity Flag
  - CF Carry Flag

セグメント

- 有効アドレス Effective Address
  - 有効addr (20-bit)
    = Segment address (16 + 4bit)
    + Offset addr　　(8086，16bitの例）
  - 例) CS = 16AD, Offset = 0100の時,
    IP = 16AD0 + 0100 = _____
- セグメントレジスター
  - DS Data Segment
  - CS Code Segment
  - ES Extra Segment
  - SS Stack Segment

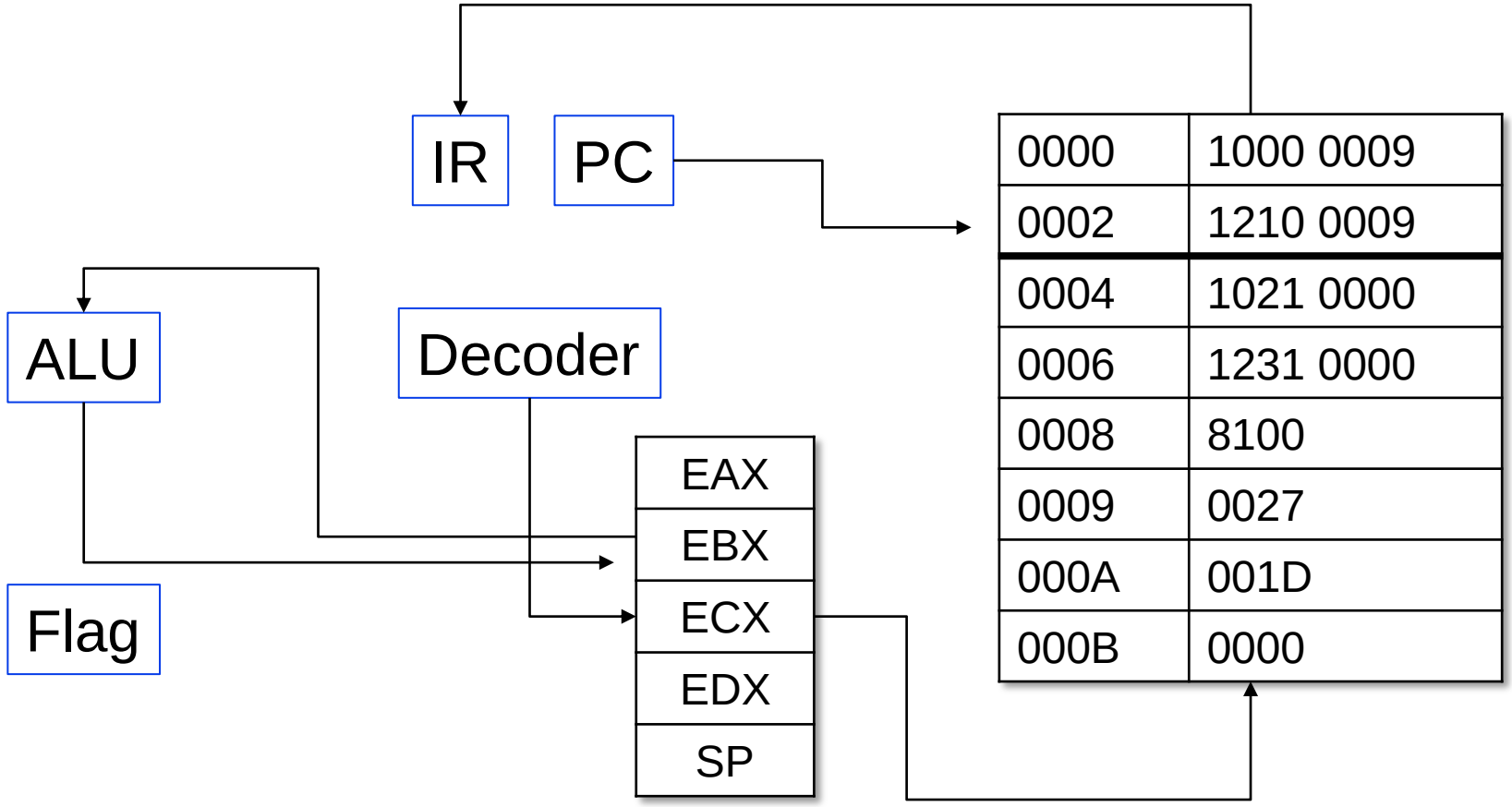
アドレス空間

CPU	アドレスバス	例	メモリ容量
8 bit	16 bit	8085, Z-80	64 KB (2^{16})
16 bit		8086	1MB (2^{20})
32 bit	32 bit	Pentium D	
64 bit	64 bit	Core 2 Duo, I 7	16EB (2^{64})

2. 動作の流れ

- 実行ステップ
  - 1. 命令読み出し (　　)
  - 2. 命令解読 (decode)
  - 3. 実行 (execute)
  - 4. 結果格納 (write-back)

命令の処理

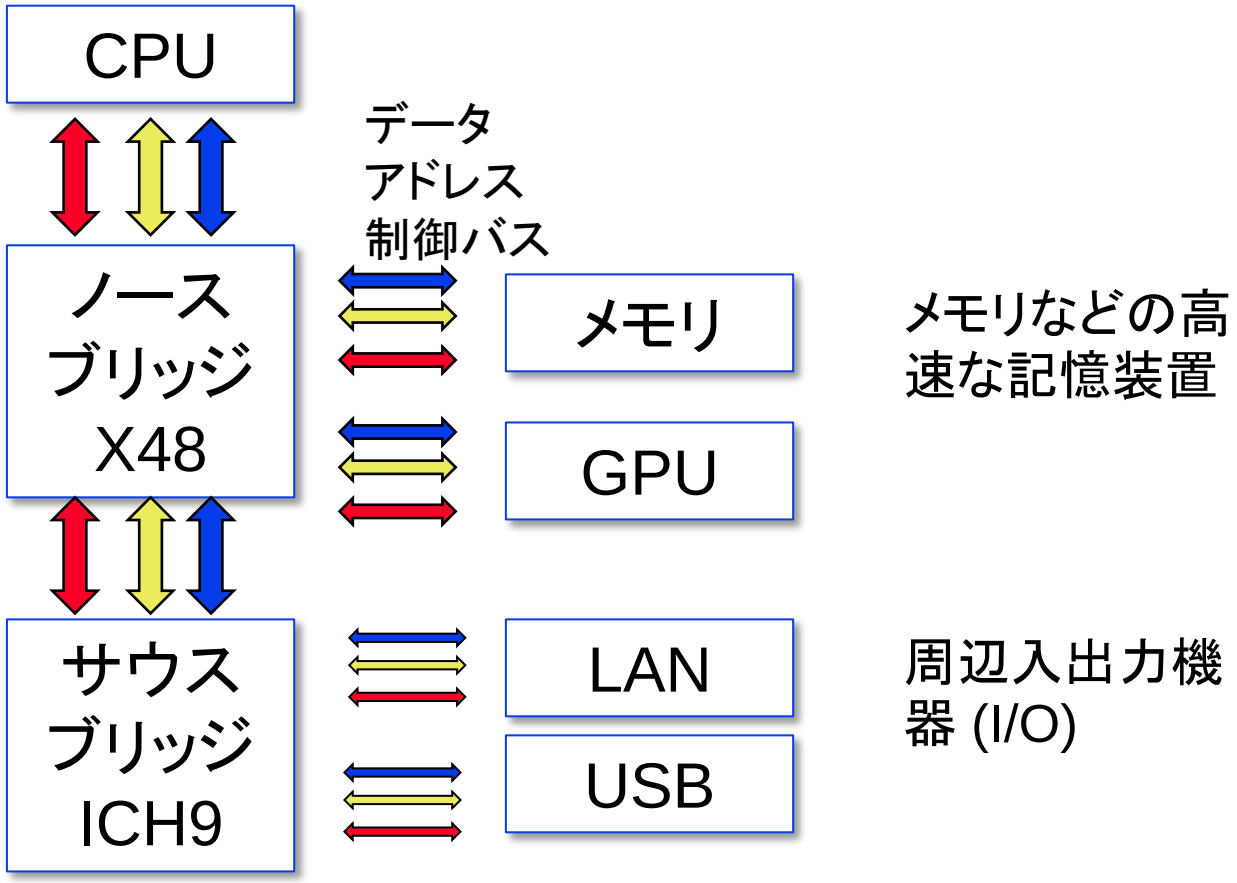

IR
PC
ALU
Decoder
Flag
EAX
EBX
ECX
EDX
SP
0000
1000 0009
0002
1210 0009
0004
1021 0000
0006
1231 0000
0008
8100
0009
0027
000A
001D
000B
0000

Bus

- Bus（8086の例）双方向データ信号線
  - アドレスバス A_0-A_{19}
  - データバス A_0-A_{15}
  - 制御バス S_0-S_7

時刻	1	2	3	4	5	6	7	8	9
S_2 S_1 S_0	100	101	000	110	100	101	000	110	100
バス	アドレス	命令		データ	アドレス	命令		データ	アドレス
内容	Fetch	read	(exe)	write	Fetch	read	(exe)	Fetch	Fetch

コントロールバス

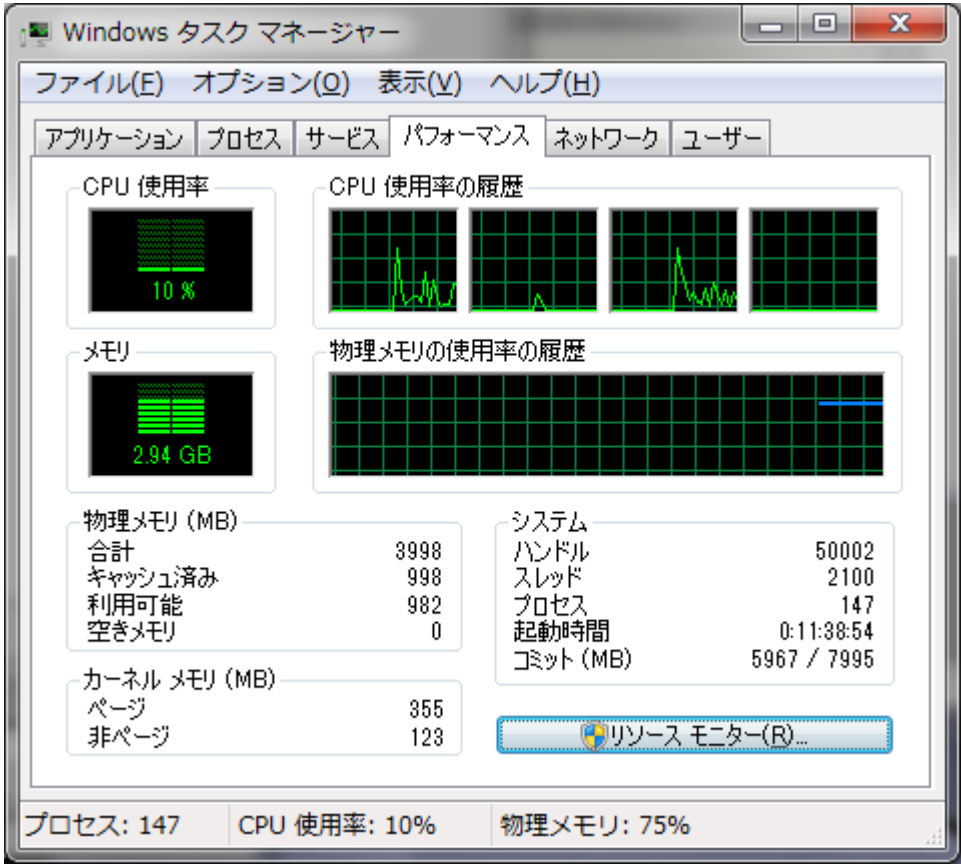

CPU
データ
アドレス
制御バス
ノース
ブリッジ
X48
メモリ
GPU
サウス
ブリッジ
ICH9
LAN
USB
メモリなどの高速な記憶装置
周辺入出力機器 (I/O)

高速化の工夫

- キャッシュメモリ
- パイプライン
- スーパースカラー
- マルチコア
- VLIW

マルチコア

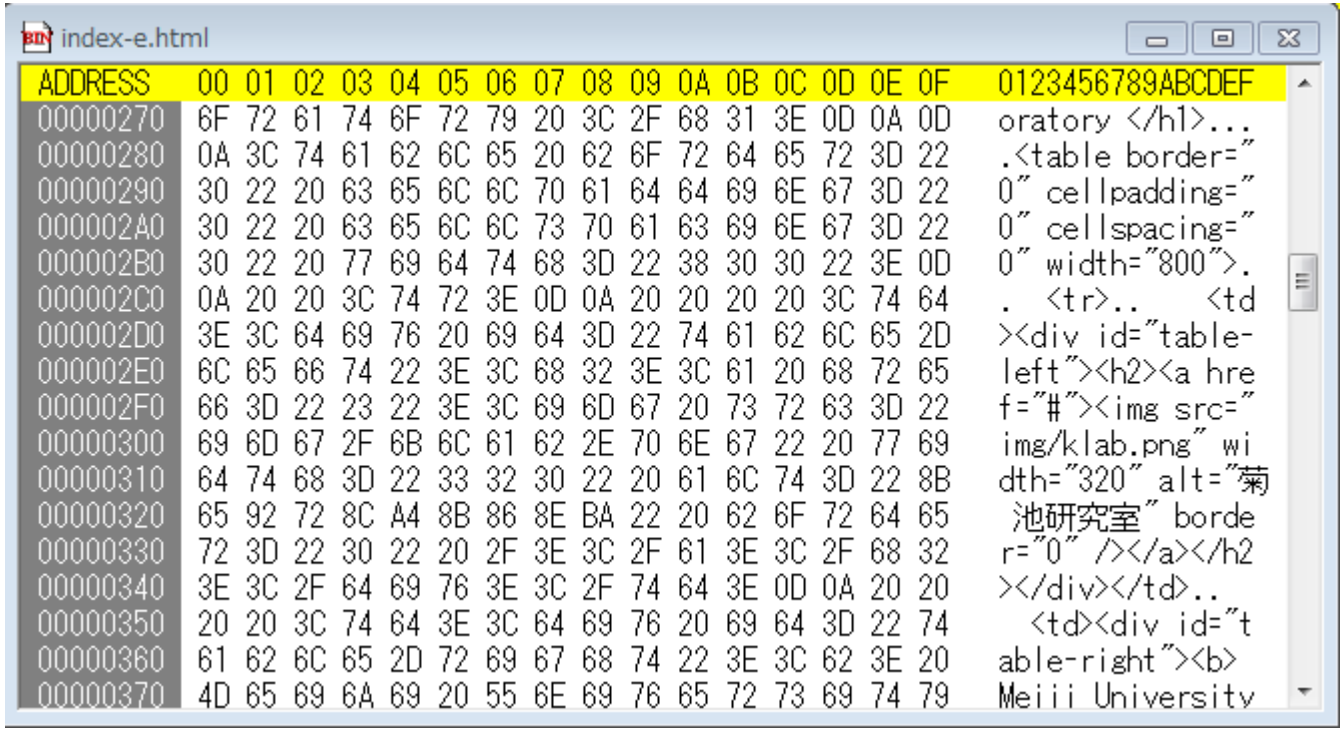
- Core
  - Fetch, Decode, Execute, Write-backなどの回路（キャッシュはない）
  - 単一のCPUに複数のコア = マルチコア
  - Core 2以降



3. メモリーダンプ

- Stirling（バイナリエディター）

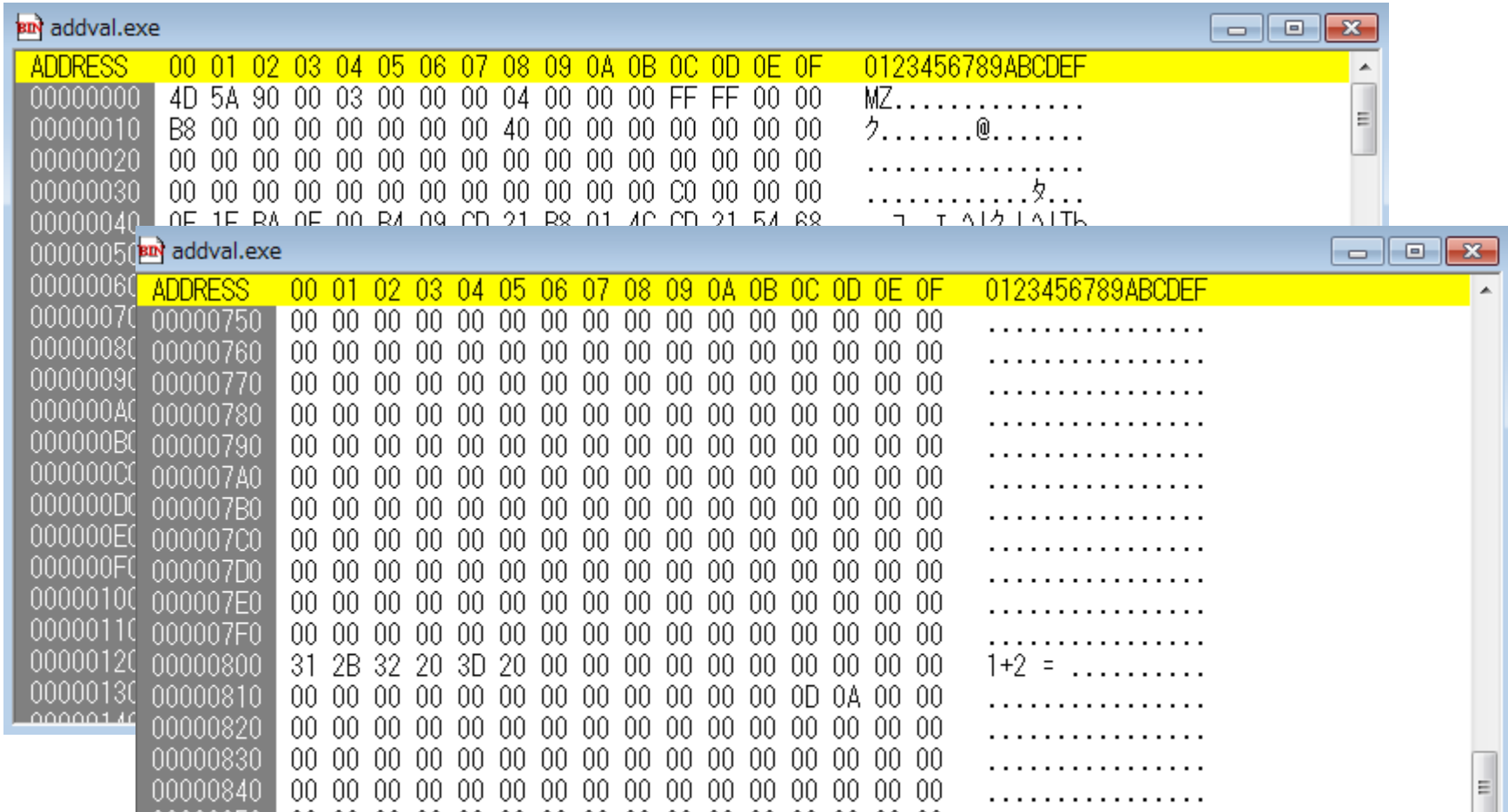
```
<table border="0"
cellpadding="0"
width="800">
  <tr>
  <td><div id="table-
left"><h2><a
href="#"><img
src="img/klab.png"
width="320" alt="菊池研
究室" border="0"
/></a></h2></div></td>
```



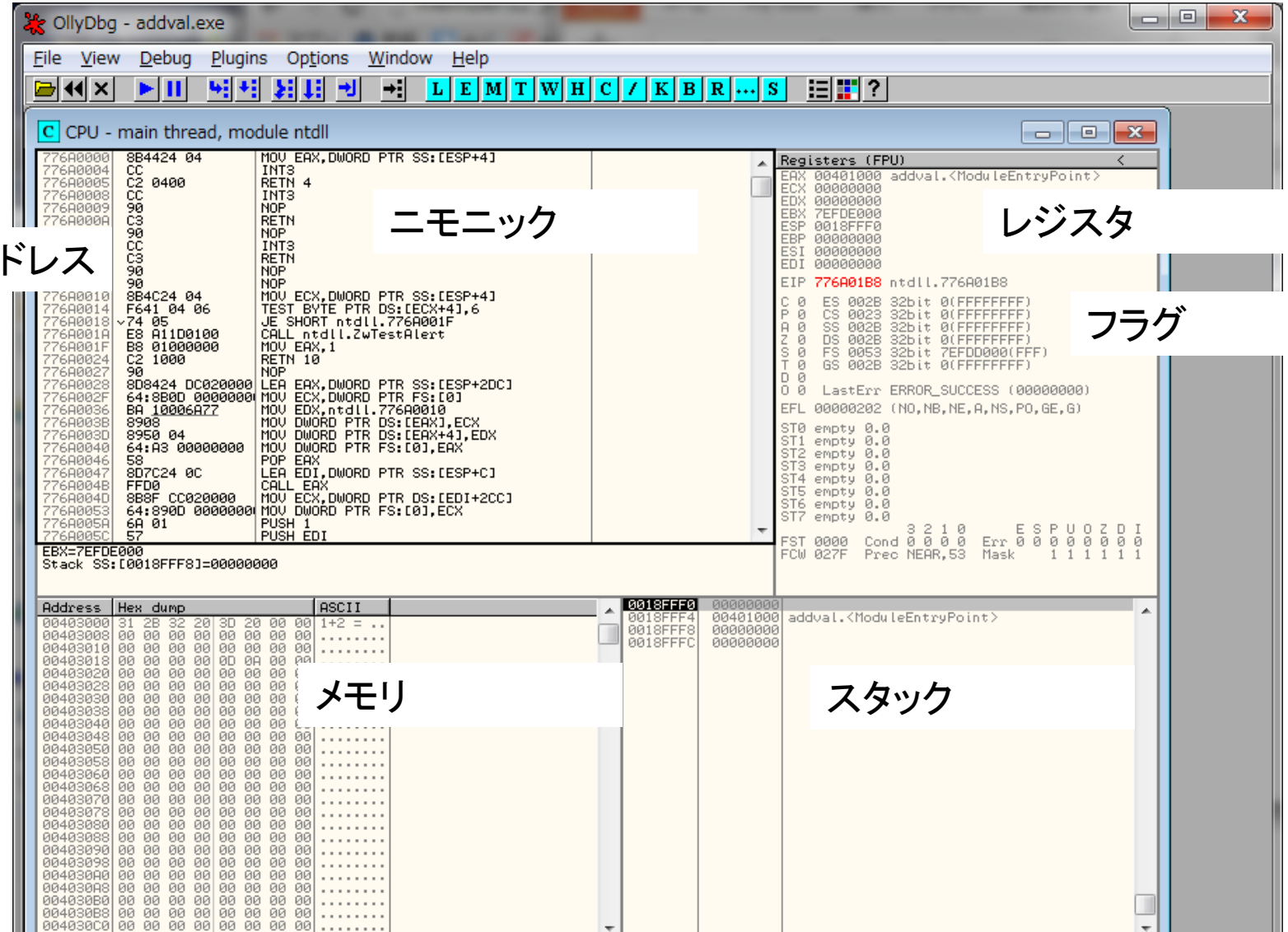
EXEファイル

```
D:¥Kikuchi¥Wrk¥maem>addval
1+2 = 3
```

- addval.exe (2,560 byte)



逆アセンブラー OllyDbg



MASM

- addval.asm

```
1.              include   D:¥masm32¥include¥masm32rt.inc
2.              .data
3.      msg db "1+2 = ", 0
4.              .code
5.      start:
6.              print OFFSET msg
7.              mov eax, 1
8.              mov ecx, 2
9.              add ecx, eax
10.             print str$(ecx)
11.             print chr$(13,10)
12.             exit
13.     end start
```

CASLとIA32の比較

種類	IA32	CASL

データ転送	**MOV** EAX, EDX
	MOV EAX, 1
	LEA EAX, 0123h
	MOV EAX, 1, EEX
		LD GR0, GR1
		LAD GR0, 1
		LAD GR0, 0123
		LD　GR0, 1, GR2

データ格納	MOV 0100h, EAX	ST　GR0, 0100

演算命令	ADD EAX, ECX
	SUB EAX, ECX
	CMP EAX, EBX
	INC EAX
		ADDA GR0, GR1
		SBUA GR1, GR2
		CPA GR0, GR1
		LAD　GR0, 1, GR0

制御，分岐命令	**JZ** 0100h **JLE** 0100h JG　0100h	JZE 0100 JMI 0100 JPL 0100

宿題

- 4章
  - 問8
  - 問9

まとめ

- インテルアーキテクチャーには，(　　)bitの8080から，(　　) core i-7までの種類がある.
- EAX, EBXなどの64-bitの（　　）と，EEX, EDXなどの（　　）を持つ.
- 機械語の実行は，命令の（　　），読み出し，実行，（　　）の主に4つのフェーズがあり，それらを並列に実行することで高速化する技術を（　　）という.
- OllyDbgなどの機械語をアセンブラ言語に変換するツールを（　　）という.