# Intel Architecture Code Analyzer について

x86/x64最適化勉強会#2 (2011/10/1)

Shiraishi Masao

# 自己紹介

- 白石匡央（msiro）
- ブログ：Coding Memorandum
  http://msirocoder.blog35.fc2.com/
- 仕事：映像Codec，トランスコーダの開発
- 趣味：競技プログラミング
- 変遷：MSX（Z80）→ SunOS（SPARC）
  →Windows(x86)

# 本日の発表について

- 静的解析ツール「Intel Architecture Code Analyzer」を取り上げます。

- Intel CPUのアーキテクチャをより深く理解できる（かもしれません）

# x86/x64最適化

- コードを速くしたい
  - 並列化
  - ワークロード最適化
    - SIMD化　→　データ構造を選ぶ
    - ボトルネックの排除

      Intelアーキテクチャは複雑

      → ボトルネック要因が簡単には分からない

      → ツールを使おう！

一般的にはプロファイラ（実行解析）を使います。

でも，本日は「静的解析ツール」を取り上げます。

# whatif.intel.com

- インテルの実験的ソフトウェアの公開の場

Performance Tuning

- Intel Performance Bottleneck Analyzer
- Intel Software Autotuning Tool
- Intel Software Tuning Agent
- Intel Architecture Code Analyzer：（IACA）
- Intel Performance Tuning Utility
- Intel Platform Modeling with Machine Learning

# IACA機能概要

- 機能概要
  - x86/x64命令がどのportで実行されるかを示す
    front-end，OOO，メモリキャッシュを考慮する
  - 各portのサイクル数を示す
  - （コードの静的解析による）スループットとレイテンシを示す
  - クリティカルパスを示す

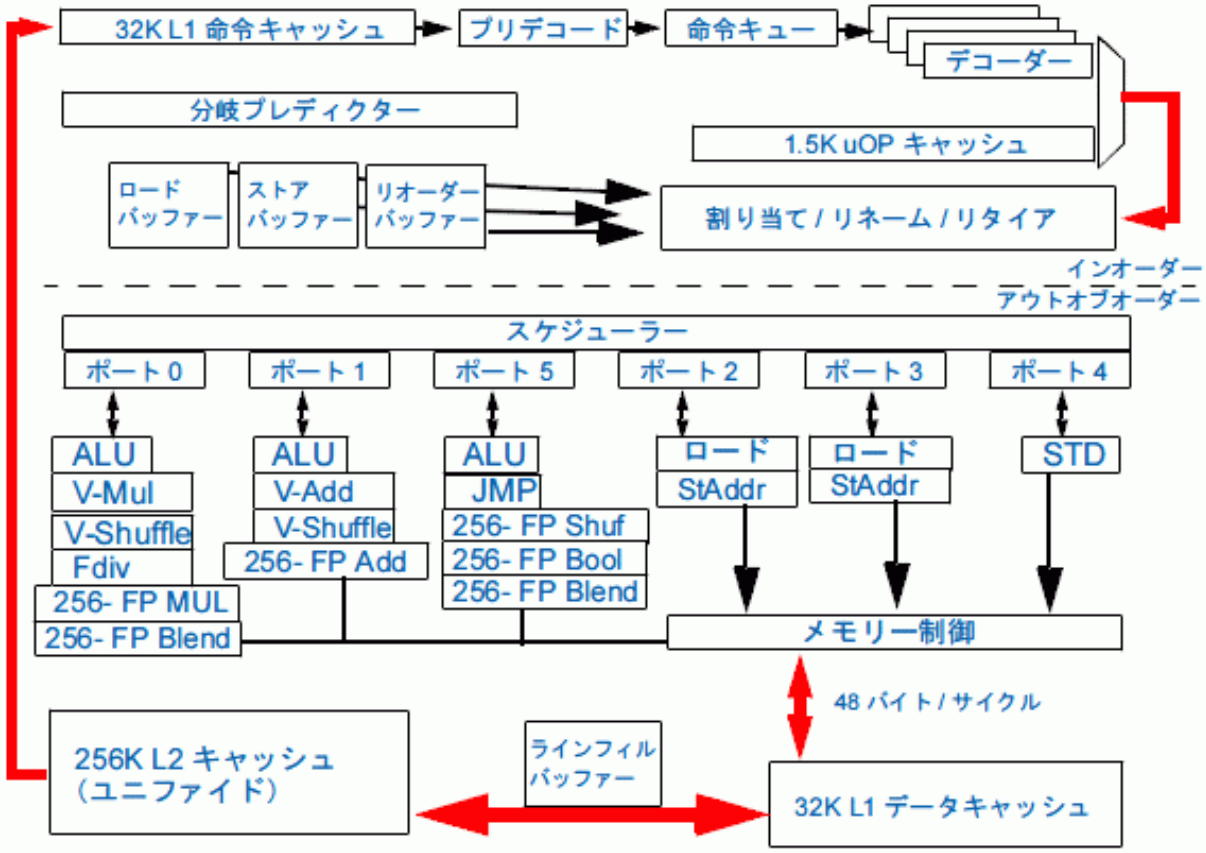
- 元々はSandy Bridgeが出る前にAVXを評価するツールだった（?）
  Ver.1.1からNehalem，Westmereをサポート

IACAはASCII出力のコマンドラインツールです。
次画面で実行結果例を示します。

```
Intel(R) Architecture Code Analyzer Version - 1.1.3
Analyzed File - ..¥test¥Release¥test.exe
Binary Format - 32Bit
Architecture  - Intel(R) AVX

Analysis Report
---------------
Total Throughput: 76 Cycles;            Throughput Bottleneck: Port0, Port1
Total number of Uops bound to ports:  340
Data Dependency Latency:    51 Cycles;  Performance Latency:    103 Cycles

Port Binding in cycles:
-------------------------------------------------
|  Port  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |
-------------------------------------------------
| Cycles | 76 |  0 | 76 | 48 | 39 | 47 | 38 | 18 | 75 |
-------------------------------------------------

| Num of |          Ports pressure in cycles          |    |
|  Uops  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |    |
---------------------------------------------------------------
|   2^   |    |    |    | X |    | 1 |    | 1 |    | CP | mov dword ptr [esp+0xc], eax
|   2^   |    |    |    | 1 |    | X |    | 1 |    |    | mov dword ptr [esp+0x8], ecx
|   1    |    |    |    | 1 |  1 | X |  X |   |    | CP | mov eax, dword ptr [esp+0xc]
|   1    |    |    |    | X |  X | 1 |  1 |   |    |    | mov edx, dword ptr [esp+0x8]
|   1    |    |    |    | 1 |  1 | X |  X |   |    |    | movdqa xmm0, xmmword ptr [eax]
|   1    |    |    |  1 |   |    |   |    |   |    |    | lea esi, ptr [0x4030b0]
|   1    |    |    |    | X |  X | 1 |  1 |   |    |    | movdqa xmm4, xmmword ptr [eax+0x20]
|   1    |    |    |  1 |   |    |   |    |   |    |    | lea ecx, ptr [0x403130]
|   1    |  1 |    |    |   |    |   |    |   |  X |    | pshuflw xmm0, xmm0, 0xd8
|   1    |  X |    |    |   |    |   |    |   |  1 |    | pshufd xmm1, xmm0, 0x0
|   2^   |    |    |  1 |  1 |  1 | X |  X |   |    |    | pmaddwd xmm1, xmmword ptr [esi]
|   1    |  1 |    |    |   |    |   |    |   |  X |    | pshufd xmm3, xmm0, 0x55
|   1    |  X |    |    |   |    |   |    |   |  1 |    | pshufhw xmm0, xmm0, 0xd8
|   2^   |    |    |  1 |  X |  X |  1 |  1 |   |    |    | pmaddwd xmm3, xmmword ptr [esi+0x20]
|   1    |  1 |    |    |   |    |   |    |   |  X |    | pshufd xmm2, xmm0, 0xaa
|   1    |  X |    |    |   |    |   |    |   |  1 |    | pshufd xmm0, xmm0, 0xff
```

# 前提知識（パイプライン）

▼SandyBridgeのパイプライン




x86/x64命令

front-end

μ OPs

スケジューラ

発行ポート
実行ユニット

「インテル® 64 アーキテクチャーおよびIA-32 アーキテクチャー最適化リファレンス・マニュアル」より

# 前提知識（発行ポート）

▼SandyBridgeの発行ポートと実行ユニット

| | ポート0 | ポート1 | ポート2 | ポート3 | ポート4 | ポート5 |
|---|---|---|---|---|---|---|
| 整数 | ALU、Shift | ALU、Fast LEA、Slow LEA、MUL | Load_Addr、Store_addr | Load_Addr Store_addr | Store_data | ALU、Shift、Branch、Fast LEA |
| SSE-Int、AVX-Int、MMX | Mul、Shift、STTNI、Int-Div、128b-Mov | ALU、Shuf、Blend、128b-Mov | | | Store_data | ALU、Shuf、Shift、Blend、128b-Mov |
| SSE-FP、AVX-FP_low | Mul、Div、Blend、256b-Mov | Add、CVT | | | Store_data | Shuf、Blend、256b-Mov |
| X87、AVX-FP_High | Mul、Div、Blend、256b-Mov | Add、CVT | | | Store_data | Shuf、Blend、256b-Mov |

「インテル® 64 アーキテクチャーおよびIA-32 アーキテクチャー最適化リファレンス・マニュアル」より

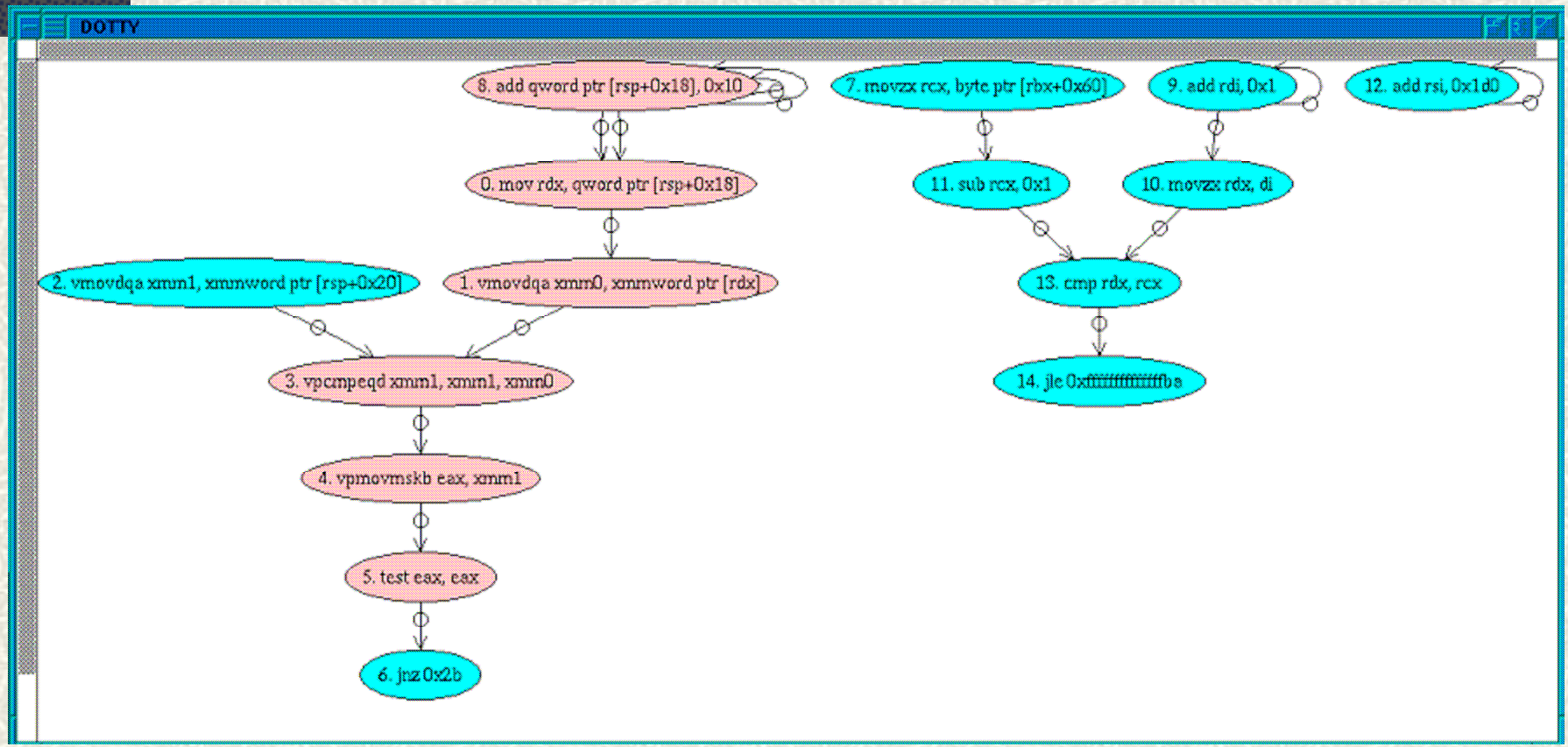
# IACAを知るきっかけ

- Intel Software Developer Day 2011（7/15）
  - 「SandyBridge向けワークロードの最適化」の中で紹介
  - ボトルネックの大部分はイベントベースの解析で検知できる
  - IACAもしくはコードの調査のみで判別できるものがある
    → 特定Portへの高い依存性の検知

次ページの画面例とともに紹介

```
iaca.exe -arch SNB -analysis THROUGHPUT -graph g.dot regspill.exe
```

# IACA画面例？




「Intel Software Developer Day インテルマイクロアーキテクチャーSandyBridge向けワークロードの最適化」より

しかしながら，現在の公開版（Rev.1.1.3）にこのような機能はない・・・

# IACAの使い方

- 解析対象のコードブロックの前後にマーカー(10byte/x86)を入れる
  - iacaMarks.h にマクロが定義される
    IACA_START：解析コードブロックの開始位置
    IACA_END　：解析コードブロックの終了位置
- コマンド引数

| | |
|---|---|
| -32 | 32ビット オブジェクト |
| -64 | 64ビット オブジェクト |
| -arch <type> | AVX, nehalem, westmere |
| -cp <type> | DATA_DEPENDENCY, PERFORMANCE |
| -include_ebx | マーカー前後のpop/push ebxを解析対象に含める |
| -mark <n> | n番目のブロックを解析する　0のとき全て |
| -o <ファイル名> | 出力ファイル名 |

# IACA出力詳細

```
Intel(R) Architecture Code Analyzer Version - 1.1.3
Analyzed File - ..¥test¥Release¥test.exe
Binary Format - 32Bit
Architecture  - Intel(R) AVX

Analysis Report
---------------
Total Throughput: 76 Cycles;            Throughput Bottleneck: Port0, Port1
Total number of Uops bound to ports:  340
Data Dependency Latency:    51 Cycles;  Performance Latency:    103 Cycles

Port Binding in cycles:
-------------------------------------------------------
|  Port  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |
-------------------------------------------------------
| Cycles | 76 |  0 | 76 | 48 | 39 | 47 | 38 | 18 | 75 |
-------------------------------------------------------

| Num of |          Ports pressure in cycles          |    |
|  Uops  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |    |
---------------------------------------------------------------
|   2^   |    |    |    |  X |    |  1 |    |  1 |    | CP | mov dword ptr [esp+0xc], eax
|   2^   |    |    |    |  1 |    |  X |    |  1 |    |    | mov dword ptr [esp+0x8], ecx
|   1    |    |    |    |  1 |  1 |  X |  X |    |    | CP | mov eax, dword ptr [esp+0xc]
|   1    |    |    |    |  X |  X |  1 |  1 |    |    |    | mov edx, dword ptr [esp+0x8]
|   1    |    |    |    |  1 |  1 |  X |  X |    |    |    | movdqa xmm0, xmmword ptr [eax]
|   1    |    |    |  1 |    |    |    |    |    |    |    | lea esi, ptr [0x4030b0]
|   1    |    |    |    |  X |  X |  1 |  1 |    |    |    | movdqa xmm4, xmmword ptr [eax+0x20]
|   1    |    |    |  1 |    |    |    |    |    |    |    | lea ecx, ptr [0x403130]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshuflw xmm0, xmm0, 0xd8
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufd xmm1, xmm0, 0x0
|   2^   |    |    |  1 |  1 |  1 |  X |  X |    |    |    | pmaddwd xmm1, xmmword ptr [esi]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshufd xmm3, xmm0, 0x55
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufhw xmm0, xmm0, 0xd8
|   2^   |    |    |  1 |  X |  X |  1 |  1 |    |    |    | pmaddwd xmm3, xmmword ptr [esi+0x20]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshufd xmm2, xmm0, 0xaa
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufd xmm0, xmm0, 0xff
```

# IACA出力詳細(サマリー)

```
Analysis Report
---------------
Total Throughput: 76 Cycles;            Throughput Bottleneck: Port0, Port1
Total number of Uops bound to ports:  340
Data Dependency Latency:    51 Cycles;  Performance Latency:    103 Cycles
```

- コードブロック全体のスループットとレイテンシ
- スループットのボトルネック
- μOP数

スループットは次に示すものの最大値である。(=ボトルネック)

- 各発行ポートのスループット
- front-end の最大スループット (= 4 μOPs/cycle)
- 除算ユニットのスループット

# IACA出力詳細(サマリー)

```
Analysis Report
---------------
Total Throughput: 76 Cycles;             Throughput Bottleneck: Port0, Port1
Total number of Uops bound to ports:  340
Data Dependency Latency:    51 Cycles;  Performance Latency:    103 Cycles
```

Data Dependecy Latency: Data Dependencyクリティカルパスの実行サイクル数
(前命令の結果を必要とする命令系列の最大パス)

Performance Latency: Performanceクリティカルパスの実行サイクル数

▼次の基準を考慮

- 前命令の実行結果に依存する命令
- front-end で遅れる命令
- ポートコンフリクトにより遅れる命令

# IACA出力詳細

```
Intel(R) Architecture Code Analyzer Version - 1.1.3
Analyzed File - ..¥test¥Release¥test.exe
Binary Format - 32Bit
Architecture  - Intel(R) AVX

Analysis Report
---------------
Total Throughput: 76 Cycles;            Throughput Bottleneck: Port0, Port1
Total number of Uops bound to ports:  340
Data Dependency Latency:    51 Cycles;  Performance Latency:    103 Cycles

Port Binding in cycles:
-----------------------------------------------------
|  Port  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |
-----------------------------------------------------
| Cycles | 76 |  0 | 76 | 48 | 39 | 47 | 38 | 18 | 75 |
-----------------------------------------------------

| Num of |          Ports pressure in cycles          |    |
|  Uops  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |    |
---------------------------------------------------------
|   2^   |    |    |    |  X |    |  1 |    |  1 |    | CP | mov dword ptr [esp+0xc], eax
|   2^   |    |    |    |  1 |    |  X |    |  1 |    |    | mov dword ptr [esp+0x8], ecx
|   1    |    |    |    |  1 |  1 |  X |  X |    |    | CP | mov eax, dword ptr [esp+0xc]
|   1    |    |    |    |  X |  X |  1 |  1 |    |    |    | mov edx, dword ptr [esp+0x8]
|   1    |    |    |    |  1 |  1 |  X |  X |    |    |    | movdqa xmm0, xmmword ptr [eax]
|   1    |    |    |  1 |    |    |    |    |    |    |    | lea esi, ptr [0x4030b0]
|   1    |    |    |    |  X |  X |  1 |  1 |    |    |    | movdqa xmm4, xmmword ptr [eax+0x20]
|   1    |    |    |  1 |    |    |    |    |    |    |    | lea ecx, ptr [0x403130]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshuflw xmm0, xmm0, 0xd8
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufd xmm1, xmm0, 0x0
|   2^   |    |    |  1 |  1 |  1 |  X |  X |    |    |    | pmaddwd xmm1, xmmword ptr [esi]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshufd xmm3, xmm0, 0x55
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufhw xmm0, xmm0, 0xd8
|   2^   |    |    |  1 |  X |  X |  1 |  1 |    |    |    | pmaddwd xmm3, xmmword ptr [esi+0x20]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshufd xmm2, xmm0, 0xaa
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufd xmm0, xmm0, 0xff
```

# IACA出力詳細(Port毎のサイクル)

```
Port Binding in cycles:
-----------------------------------------------------
|  Port  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |
-----------------------------------------------------
| Cycles | 76 |  0 | 76 | 48 | 39 | 47 | 38 | 18 | 75 |
-----------------------------------------------------
```

Port0, 2, 3には, regular pipe と secondary pipe がある。

Port0：Divider Pipe
第1サイクルは双方ビジー。次サイクルではPort0が使用可能。
Divider pipeは, 除算実行が終わるまでビジー

Port2,3：Address Generate Unit(AGU)
256bitロードは2サイクルPortビジーであるが,
AGUは1サイクル後にフリーとなりstore address generation可

# IACA出力詳細(Port毎のサイクル)

- CPUによる違い
  「AP-945 SSE2を使用した逆離散コサイン変換」

-arch AVX

```
Port Binding in cycles:
-----------------------------------------------------
|  Port  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |
-----------------------------------------------------
| Cycles | 76 |  0 | 76 | 48 | 39 | 47 | 38 | 18 | 75 |
-----------------------------------------------------
```

-arch westmere / nehalem

```
Port Binding in cycles:
-----------------------------------------------------
|  Port  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |
-----------------------------------------------------
| Cycles | 76 |  0 | 76 | 77 | 77 | 18 |  0 | 18 | 75 |
-----------------------------------------------------
```

# SandyBridge向け最適化

- 2個/サイクルのロードを実行できる

▼高速化前

```
loop_start:
  paddd xmm0, [rsi]
  paddd xmm0, [rsi+16]
  paddd xmm0, [rsi+32]
  paddd xmm0, [rsi+48]
  paddd xmm0, [rsi+64]
  paddd xmm0, [rsi+80]
  paddd xmm0, [rsi+96]
  paddd xmm0, [rsi+112]
```

▼高速化後

```
loop_start:
  paddd xmm0, [rsi]
  paddd xmm1, [rsi+16]
  paddd xmm0, [rsi+32]
  paddd xmm1, [rsi+48]
  paddd xmm0, [rsi+64]
  paddd xmm1, [rsi+80]
  paddd xmm0, [rsi+96]
  paddd xmm1, [rsi+112]
```

「Intel Software Developer Day
インテルマイクロアーキテクチャーSandyBridge向けワークロードの最適化」によれば，
1.65倍早くなる。（ループ処理のオーバヘッドで2倍にならない）

# IACA出力詳細

```
Intel(R) Architecture Code Analyzer Version - 1.1.3
Analyzed File - ..¥test¥Release¥test.exe
Binary Format - 32Bit
Architecture  - Intel(R) AVX

Analysis Report
---------------
Total Throughput: 76 Cycles;            Throughput Bottleneck: Port0, Port1
Total number of Uops bound to ports:  340
Data Dependency Latency:    51 Cycles;  Performance Latency:    103 Cycles

Port Binding in cycles:
-------------------------------------------------------
|  Port  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |
-------------------------------------------------------
| Cycles | 76 |  0 | 76 | 48 | 39 | 47 | 38 | 18 | 75 |
-------------------------------------------------------

| Num of |          Ports pressure in cycles          |    |
|  Uops  |  0 - DV |  1 |  2 -  D |  3 -  D |  4 |  5 |    |
---------------------------------------------------------------
|   2^   |    |    |    |  X |    |  1 |    |  1 |    | CP | mov dword ptr [esp+0xc], eax
|   2^   |    |    |    |  1 |    |  X |    |  1 |    |    | mov dword ptr [esp+0x8], ecx
|   1    |    |    |    |  1 |  1 |  X |  X |    |    | CP | mov eax, dword ptr [esp+0xc]
|   1    |    |    |    |  X |  X |  1 |  1 |    |    |    | mov edx, dword ptr [esp+0x8]
|   1    |    |    |    |  1 |  1 |  X |  X |    |    |    | movdqa xmm0, xmmword ptr [eax]
|   1    |    |    |  1 |    |    |    |    |    |    |    | lea esi, ptr [0x4030b0]
|   1    |    |    |    |  X |  X |  1 |  1 |    |    |    | movdqa xmm4, xmmword ptr [eax+0x20]
|   1    |    |    |  1 |    |    |    |    |    |    |    | lea ecx, ptr [0x403130]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshuflw xmm0, xmm0, 0xd8
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufd xmm1, xmm0, 0x0
|   2^   |    |    |  1 |  1 |  1 |  X |  X |    |    |    | pmaddwd xmm1, xmmword ptr [esi]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshufd xmm3, xmm0, 0x55
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufhw xmm0, xmm0, 0xd8
|   2^   |    |    |  1 |  X |  X |  1 |  1 |    |    |    | pmaddwd xmm3, xmmword ptr [esi+0x20]
|   1    |  1 |    |    |    |    |    |    |    |  X |    | pshufd xmm2, xmm0, 0xaa
|   1    |  X |    |    |    |    |    |    |    |  1 |    | pshufd xmm0, xmm0, 0xff
```

# IACA出力詳細(命令詳細)

| Num of Uops | Ports pressure in cycles | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 - DV | 1 | 2 - D | | 3 - D | | 4 | 5 | | |
| 2^ | | | | X | | 1 | | 1 | | CP | mov dword ptr [esp+0xc], eax |
| 2^ | | | | 1 | | X | | 1 | | | mov dword ptr [esp+0x8], ecx |
| 1 | | | | 1 | 1 | X | X | | | CP | mov eax, dword ptr [esp+0xc] |
| 1 | | | | X | X | 1 | 1 | | | | mov edx, dword ptr [esp+0x8] |
| 1 | | | | 1 | 1 | X | X | | | | movdqa xmm0, xmmword ptr [eax] |
| 1 | | | | 1 | 1 | X | X | | | CP | mov eax, dword ptr [ebp-0x8] |
| ! | ! | ! | ! | ! | ! | ! | ! | ! | ! | | cdq |
| 1 | 1 | | X | | | | | | X | | mov ecx, 0x3 |
| ! | ! | ! | ! | ! | ! | ! | ! | ! | ! | | idiv ecx |

Num of Uops：μOp数

X：このportでも実行可能であることを示す

CP：クリティカルパス

！：サポート対象外の命令

# IACA出力詳細(命令詳細)

| Num of Uops | Ports pressure in cycles 0 - DV | 1 | 2 - D | 3 - D | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|
| 0* | | | | | | | | xor eax, eax |
| 1 | 1 | | X | | | | X | | nop |
| 1 | X | | 1 | | | | X | | add ecx, eax |
| 1 | X | | X | | | | 1 | CP | add eax, 0x1 |
| 1 | X | | X | | | | 1 | CP | cmp eax, 0x2710 |
| 0F | | | | | | | | | jl 0xffffffff4 |

＊: portに割り当てられない命令

- zero idiom
  - XOR REG, REG
  - SUB REG, REG
  - PXOR/VPXOR XMMREG, XMMREG
  - PSUBB/W/D/Q XMMREG, XMMREG
  - VPSUBB/W/D/Q XMMREG, XMMREG
  - XORPS/PD XMMREG, XMMREG
  - VXORPS/PD YMMREG, YMMREG
- NOP
- VZEROUPPER
- FXCHG

# IACA出力詳細(命令詳細)

```
| Num of |          Ports pressure in cycles          |    |
|  Uops  | 0 - DV | 1 | 2 -  D | 3 -  D | 4 | 5 |    |
---------------------------------------------------------------------------
|   1    |        |   |  1 |  1 |  X |  X |   |   |    | movzx eax, byte ptr [ebp-0x11]
|   1    |  X     |   |  X |    |    |    |   |   | 1  | test eax, eax
|   0F   |        |   |    |    |    |    |   |   |    | jz 0x9
|   2^   |        |   |  1 |    |  X |    | 1 |   |    | mov dword ptr [ebp-0x8], 0xa
```

F : マクロフュージョン
　2つの命令を単一μOPにマージ

表 3-1　インテル® マイクロアーキテクチャー Sandy Bridge におけるマクロフュージョン可能な命令

| 命令 | TEST | AND | CMP | ADD | SUB | INC | DEC |
|---|---|---|---|---|---|---|---|
| JO/JNO | Y | Y | N | N | N | N | N |
| JC/JB/JAE/JNB | Y | Y | Y | Y | Y | N | N |
| JE/JZ/JNE/JNZ | Y | Y | Y | Y | Y | Y | Y |
| JNA/JBE/JA/JNBE | Y | Y | Y | Y | Y | N | N |
| JS/JNS/JP/JPE/JNP/JPO | Y | Y | N | N | N | N | N |
| JL/JNGE/JGE/JNL/JLE/JNG/JG/JNLE | Y | Y | Y | Y | Y | Y | Y |

「インテル® 64 アーキテクチャーおよびIA-32 アーキテクチャー最適化リファレンス・マニュアル」より

# IACA出力詳細(命令詳細)

```
| Num of |          Ports pressure in cycles          |    |
|  Uops  | 0 - DV | 1 | 2 -  D | 3 -  D | 4 | 5 |    |
---------------------------------------------------------------------
|   1    |        |   |  1 |  1 |  X |  X |   |   |    | movzx eax, byte ptr [ebp-0x11]
|   1    |  X     |   |  X |    |    |    |   |   | 1  | test eax, eax
|   0F   |        |   |    |    |    |    |   |   |    | jz 0x9
|   2^   |        |   |  1 |    |  X |    | 1 |   |    | mov dword ptr [ebp-0x8], 0xa
```

^ : マイクロフュージョン

複数のμOPを単一のμOPに纏める。

例 : ロード + op

```
ADDPS XMM9, OWORD PTR [RSP+40]
FADD  DOUBLE PTR [RDI+RSI*8]
XOR   RAX, QWORD PTR [RBP+32]
```

# IACA出力詳細(命令詳細)

| Num of Uops | Ports pressure in cycles 0 - DV | 1 | 2 - D | 3 - D | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|
| 2 | | 1 | X | X | 1 | 2 | | CP | vaddpd ymm11, ymm10, ymmword ptr [rsi+rax*8] |
| 2 | | | 1 | | X | | 2 | CP | vmovaps ymmword ptr [rcx+rax*8], ymm12 |
| 1 | 1 | | | | | | | CP | vmulpd ymm14, ymm3, ymm11 |
| 1@ | | | 1 | | | | | | addps xmm0, xmm1 |
| 1 | 1 | | | | | | | | vmulpd ymm12, ymm2, ymm14 |
| 3 | 2 | 42 | | | | | 1 | CP | vsqrtpd ymm4, ymm5 |

@： AVX-256 に SSE コードが続くケース
→ ペナルティが発生

- SSEの代わりにAVX-128命令を使う。
- SSEコードの直前にVZEROUPPERを入れる。

# まとめ

- IACAでは，スーパースカラ実行の効率を評価することができる。（サマリ画面）
- コードのどこを変えれば良いかは分かり難い。
- Intel Software Developer Dayで紹介された画面機能が公開されることを期待したい。