

スーパースケラ技法を用いた i960CA アーキテクチャ

山田 功

インテルジャパン (株)

インテル・コーポレーション (本社: アメリカ合衆国カリフォルニア州サンタクララ市) は, 1989年秋に商用では初めてのスーパースケラ技法を用いたマイクロプロセッサ i960 CA を発表した。このスーパースケラ技法とは, 1プロセッサ・クロック・サイクルにおいて複数の独立した命令のディスパッチ及び実行を行うことのできる技法をいう。この結果, この i960 CA では多くのアルゴリズムでの演算に対して, 1クロック・サイクルにおいて連続で2命令の処理を行うことが可能となる。すなわち i960 CA では, ネイティブ MIPS として 66MIPS の性能を 33MHz 時に達成することができる。本論文においては, この内部アーキテクチャについて中心に述べるが, 簡単にこの技術のもつ影響についても考察する。

The i960CA Architecture of SuperScalar Implementation

Isao Yamada

Intel Japan K. K.

5-6 Tokodai, Tsukuba-shi, Ibaragi, 300-26, Japan

intel corporetion (Santa Clara, California, U. S. A.) announced i960 CA, the first SuperScalar implemented processor in the market, in fall of 1989. The SuperScalar technology is the capable technology of the dispatch and execution of multiple independent instructions during each processor clock cycle. The i960 CA allows an sustained execution rate of up to two instructions per clock for many algorithms. So, i960 CA achieves 66 natives MIPS at a pocessor clock rate of 33MHz. This paper discusses mainly the micro-architecture, and also refers the effect of this technology.

1. 導入

インテルの i960 ファミリは、主に組み込み型市場を目的とした高性能マイクロプロセッサである。この i960 ファミリのアーキテクチャは1980年代に、2つの大きな目的にそって開発された。1つは、高性能を達成するためにメインフレーム・コンピュータの技術及び RISC アーキテクチャの研究技術をうまく採用することである。2つめは、幅広い組み込み型市場のニーズをみたすため、10~100MIPS 超までの性能レンジを同一アーキテクチャでみたせるといことである。この目的にそい、インテルでは1988年最初の i960 ファミリ製品、80960KA, 80960KB を発表した。これらは、本講で述べる i960 CA とはオブジェクト・コードでコンパチブルとなっている。

2. アーキテクチャの発展

並列処理の手法は、メインフレーム・コンピュータにおいては古くから実現されていたが、マイクロプロセッサではごく最近まで実現しなかった。これは、主にプロセス技術に依存し、多くのトランジスタを集積することが商業上難しかったことによる。図1に今までの手法と主なプロセッサ、時期をまとめた。

手法	主なプロセッサ	時期
簡単な制御	4004, 8008	1971-75
初期 CISC	8086, 68000	1978-81
パイプラインCISC	80286, 386™	1982-85
RISC	MIPS, SPARC	1985-87
ハイブリッドRISC	i960KA / KB	1988
VLIW	i860™	1989
スーパースケラ	i960 CA	1989

図-1 マイクロプロセッサのアーキテクチャの発展

現在、32ビット・マイクロプロセッサの多くは、パイプライン CISC が主として用いられている。命令の実行は、例えばフェッチーデコード実行のように3段、もしくは、それ以上の段数に分けられる。このとき、CISC の手法では1つのステージには1つの命令しか存在しない。例えば、独立した実行部が複数、存在することはない。

スーパースケラ技法は、一般的な RISC もしくは、VLIW 技法と多くの共通点をもつ。例えば VLIW と同様に、1クロックサイクルで多くの実行部が異なった命令を処理する。また RISC と同様に、多くの命令の処理は1クロックで行われる。

3. i960 アーキテクチャ

ここでは簡単な紹介だけとする。詳細は、インテルジャパン発行のユーザーズ・マニュアル類もしくは、マイヤーズ氏ら著の本を参照されたい。

3-1 命令セット: i960 アーキテクチャは、前にも述べたように RISC の命令体系にのっとり3オペランド命令を用い、ロード/ストア・アーキテクチャとなっている。標準的な算術論理演算やメモリアクセス命令が24種類あり、他に分岐命令やコール、リターン命令などが16種類ある。

全ての i960 の命令は3つの大きなカテゴリーに分けられる。まず、レジスタを参照する多くの命令が REG 命令群である。次にメモリへのロード/ストアを行う命令が、MEM 命令群である。最後は、分岐命令、CALL, RETURN を行う CTRL 命令群である。これらの命令は、ニーモニックの上位2ビットがそれぞれ、01, 00, 10 となっており、容易に判別される。

3-2 レジスタ・アーキテクチャ: i960 は32の汎用32ビット・レジスタをもつ。そのうち16つは、グローバル・レジスタとなり、g0~g15で記される。残りの16つは、ローカル・レジスタとなり、r0~r15と記される。このローカル・レジスタは、この16組で1セット(フレーム)となり数個のキャッシュをもつ。このフレームは、サブルーチンにおいて自動的

386™, i860™ はインテルの登録商標です。

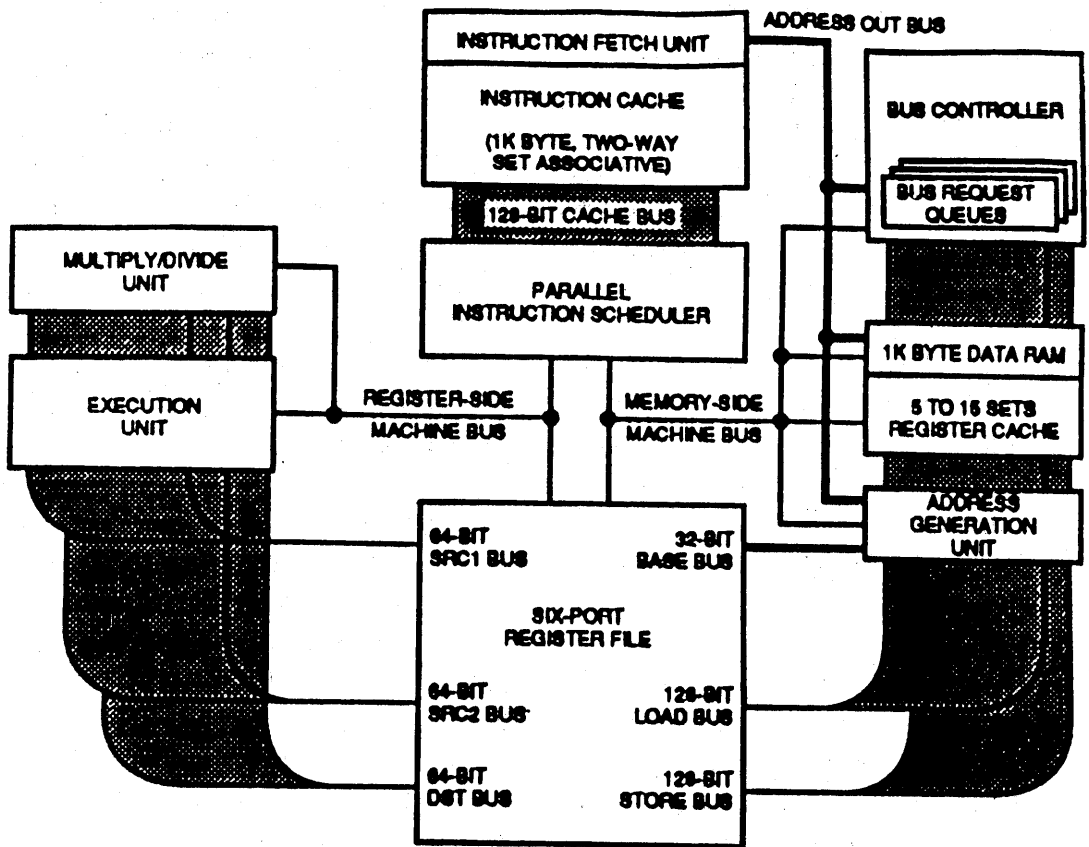


図-2 i960 CA マイクロアーキテクチャ

に変更が行われユーザーはその変更をプログラム上に指定することはない。

4. i960 CA の概要

i960 CA マイクロプロセッサの概要を列挙する。

- スーパースケラ技法を採用したプロセッサ
- 1K バイトの命令キャッシュを内蔵
- 1. 5K バイトのSRAMを内蔵
- 4チャンネルのDMA コントローラを内蔵
- 割込みコントローラを内蔵
- 外部バスは、パイプラインド・バースト・バスでアドレス/データは分離

- CHMOS-IV**, 1ミクロン CMOS テクノロジーを採用
- ダイサイズは, 9.8mm×14.6mm
- トランジスタ数は約575,000

5. i960 CA マイクロアーキテクチャ

i960CAのマイクロアーキテクチャを図2に示す。この中の大きなブロックとして、命令スケジューラ、6つのポートをもつレジスタ・ファイ

**CHMOS-IVはインテルのプロセス技術であり登録された名称です。

ル、複数の並列に処理が可能な実行ユニットが存在する。各ユニットの簡単な特徴を述べる。

5-1 6つのポートをもつレジスタ・ファイル：レジスタ・ファイルは REG 命令用の2つのソース用64ビットポートと1つのディスティネーション用64ビットポートをもつ。また、MEM 命令、CTRL 命令用の32ビット・ベース・レジスタ・アクセス用ポートと128ビットのロード及びストア用ポートをそれぞれもつ。このレジスタ・ファイルは、レジスタが使用中であることを示すスコアボーディング手法により、複数のアクセスが同時におきてもレジスタの整合性が保たれるようになっている。

5-2 命令シーケンサ：命令シーケンサは、命令キャッシュなどから命令を取り込み、REG 側もしくは MEM 側に命令をディスパッチすると同時に内部でブランチ命令の実行を行う。すなわち、最大3命令を1クロックの間にディスパッチすることができる。

命令シーケンサは命令のディスパッチの他にも命令が並列に行えるかどうか独立性をレジスタ・スコアボーディングという手法によって判断する。この手法では、今ディスパッチする命令で使用されているレジスタが、前にディスパッチされた2クロック以上の命令で使用されているかどうかの検証を行っている。ここで、2クロック以上の命令とは、乗除算、ロード等の命令である。例えば、ロード命令の実行中でまだ値が確定していないレジスタが前の命令で使用されるときは、この命令はディスパッチされない。i960 CAではこの他に、ファンクション・ユニット・スコアボーディングとコンディションコード・スコアボーディングの2つが使われている。ファンクション・ユニット・スコアボーディングとは、ある実行ユニットがまだ前の命令で実行中の場合、次の命令をディスパッチしない機能であり、i960 CAでは乗除算器でのみ適用されている。コンディションコード・スコアボーディングとは、i960 CAの分岐予測と関係がある。分岐予測が行われた場合、この予測の正否により、レジスタに値を書くなどの最終実行を決定する必要がある。この

ため、命令にスコアボーディングする必要が生じる。なお、i960CAで行う分岐予測とは、あらかじめ命令に分岐がどちらがわに行く可能性が多いかを明示することによって行われる。この方法では、約85%当たっていることが報告されている。

5-3 命令フェッチと命令キャッシュ：i960CAでは全てのクロックで最大3もしくは4命令をオンチップ命令キャッシュ、プリフェッチバッファ、マイクロコードRAMのいずれかからフェッチし、命令シーケンサへもってくる。マイクロコードROMは、i960 CAの一部の複雑な命令などのために使われており、これにより、コード密度、実行速度に貢献している。

5-4 実行ユニット：実行ユニットは REG 命令を行う整数実行ユニット (EU) と乗除算ユニット (MDU) がある。EUでは算術論理命令を行い、全て1クロックで実行する。MDUでは、乗算を4クロック、除算を34クロックで実行する。MEMサイドにある実行ユニットは、アドレス生成ユニット (AGU) である。しかし、MEMサイドには、この他にローカル・レジスタ・キャッシュ、オンチップSRAM、周辺部としてDMAユニット、バス・コントロール・ロジック (BCL) が接続されている。i960CAのアドレス生成ユニットは非常に高性能なものである。図-3にi960のアドレッシング・モードを記す。これだけの多様なモードにより、EUで行われている幾つかの命令をMEMサイドのアドレス生成ユニットで実行することが可能となる。この対応表を図-4にまとめる。これにより、並列処理の度合が上昇する。

5-5 オンチップSRAM：i960CAは約1.5KバイトのSRAMを内蔵しており、このうちの一部分は、5~15セットのローカル・レジスタ・キャッシュに使われる。また、残りの領域はNMI及び他の割込みベクタや、内部高速メモリとして使用される。SRAMは、128ビット幅のバスで接続されており、高速なデータ転送を可能としている。

6. 内蔵ペリフェラル・ユニット

Instruction	Context
ld 0x1230, g1	12bit disp
ld -foo, g1	32bit disp
ld (g0), g1	reg. indirect
ld 12 (fp), g1	reg. +offset
ld __foo [g4*16], g1	Scaled index
ld __foo (g0)[g1*4], g2	base+Scale+disp.

図-3 i960 アドレッシング・モード

REG - inst.	MEM - inst.
mov g1, g2	lda (g1), g2
addo g1, g2, g3	lda (g1) [g2*1], g3
shlo 2, g2, g3	lda [g2*4], g3
subo 1, g2, g3	lda -1 (g2), g3
lda 0x12345, g3	lda 0x12345 (g4), g5
addo g3, g4, g5	

図-4 EU / AGU 同一内容命令

i960 CA は高性能であると同時に、重要な周辺機能も内蔵している。

6-1 バス・コントローラ: i960CA のバス・コントローラはメモリマップドI/O領域を16に分割し、それぞれのバスの性格を定めることができる。ここでの設定内容は、バス幅 (8, 16, 32ビット) インターフェース・サイクル (バースト, バイブライン), ウェイト・ステート, バイト・オーダーリング (リトル・エンディアンかビッグ・エンディアン) である。外部バスは最大132MByte/秒の転送能力を持つ。また、データ・ストア時は、ライト・バッファが内蔵されていることにより、大抵1クロックのバッファへの書き込みで実行が終了する。この結果、低速な外部メモリと高速な内部動作がデカップリングされることになる。

6-2 DMA コントローラ・ユニット: i960CA は、4チャンネルの DAM コントローラを内蔵している。DMA は、ブロック・モードとダイヤモンド・

モードを持つ。また、データ幅、アラインメント、バイト・オーダーリング等の機能を持ち、フライバイ・トランスファーをサポートする。最大59Mバイト/秒の転送能をもつ。

6-3 割込みコントローラ・ユニット: i960CA は NMI の他に最大248ベクタを31プライオリティ・レベルでもつ。外部割込みはNMI 及び8本の割込みピンにより駆動される。i960CA の NMI は標準では700ns, 他の外部ピンでは1µsのレイテンシーである。

7. i960CA の性能

i960CA は、最大連続実行2命令/クロックの性能を実行できる。多くのアプリケーションは、実行の殆んどが限られたインナー・ループである場合が多く、命令キャッシュに2度め以降はヒットし、このときは2命令/クロックに近い性能が達成できる。図-5に、スタンフォード・ベンチマークから行列式の乗算の例の一部を示す。この場合、ループ内は2命令/クロックの性能を実行している。

8. i960 CA によるシステム設計への貢献

一般にシステム上での CPU の性能は、次の幾つかによって規定される。1. 命令当りの実行に必要なクロック数 2. 命令におけるコード密度 3. 命令及びデータをアクセスする際の転送速度、など。バランスのよい汎用プロセッサは、一般的なアプリケーションにおいてこのうちのどれにもよい結果を持つことが必要である。また、コストに厳しいシステムにおいては、クロック速度、メモリアクセスタイム、バンド幅などが重要な要素となっている。i960CA は、これらの点ですぐれた特性を示すマイクロプロセッサである。まず、命令実行速度は、多くの場合約1.2命令/クロックで実行し、重要なループ内では、前に示したように2命令/クロックとなりうる。次に、重要なアドレッシングモードやマイクロコー

	for j: = 1 to J do		
	for i: = 1 to I do		
	for K: = 1 to K do # The inner loop		
	C(i,j) := C(i,j) + A(i,k)*B(k,j);		
	j: mulo	g6,g8,g8	# j_loop

	i: mov		# i_loop

1	k: addo	g6,g8,g8	# Cij+ = mult123
1	ld	(g1),g3	# Aik1
2	addo	g1,r3,g1	# next A(i,k)col
2	ld	(g2),g4	# Bkj1
3	mulo	r13,r15,g5	
3	lda	4(g2),g2	# next B(k,j)
4	cmpinci	r5,r8,r8	# k = K_size?;inc k
4	bi.f	e	
5	sddo	g7,g8,g8	# Cij+ = mult3
5	ld	(g1),r12	# Aik2
6	addo	g1,r3,g1	# next A(i,k)col
6	ld	(g2),r14	# Bkj2
7	mulo	g3,g4,g6	
7	lda	4(g2),g2	# next B(k,j)
8	cmpinci	r5,r8,r8	# k = K_size?;inc k
8	bi.f	e	
9	addo	g5,g8,g8	# Cij+ = mult1
9	ld	(g1),r13	# Aik3
10	addo	g1,r3,g1	# next A(i,k)col
10	ld	(g2),r15	# Bkj3
11	mulo	r12,r14,g7	
11	lda	4(g2),g2	# next B(k,j)
12	cmpinci	r5,r8,r8	# k = K_size?;k
12	bg.t	k	
	e: cmpinci		# i=I?;inc i

	bg.t	i	
	cmpinci		# j=J?;inc j
	bg.t	j	

図-5 行列乗算ルーチン

ドROMの内蔵により、一般的なRISCアーキテクチャに比べ、コード密度の上昇を達成した。最後に、現在の33MHz動作においては、バーストモードを採用することにより一般的なDRAMを用いても少ないウェスト・ステートで実行することが可能である。くり返し述べることであるが、並列の命令実行はプロセッサ内部により達成され、VLIWのように命令への規定はない。命令コード密度は、i960アーキテクチャに渡り同一であり、並列実行処理ユニットが増えてもVLIWのように命令長をかえる必要はない。

さらに、バリエーション機能の内蔵や、容易なバス・インターフェース、比較的低い周波数の使用

により、システムの設計は同様の性能を達成する他のRISCプロセッサに比べ、格段に安くまた容易に可能となることであろう。

9. まとめ

i960 CA は1クロックで複数の命令のディスパッチ、及び実行が行える最初の汎用マイクロプロセッサである。これにより一般的なRISCプロセッサに比べ、2倍以上の性能が達成できる。この性能を最大限に引き出すために、多くの技術が用いられた。分岐予測、レジスタ・セットとローカル・レジスタ・キャッシュ、命令の分類化 (REG, MEM, CTRL命令)、スコアボーディング機能などである。

また、安価なシステムと高い転送能力の両立を可能としたバス・コントローラ、DMAコントローラ、割込みコントローラがあげられる。

10. リファレンス

80960 CA User's Manual, インテルジャパン 1989

80960KB ハードウェア・リファレンス・マニュアル, インテルジャパン 1989

80960 KB Programmer's Reference Manual, インテルジャパン 1989

G. Myers & D. Budde, The 80960 Microprocessor Architecture, Wiley Interscience, New York, NY, 1988

G. Hinton, "80960 Next Generation", Proceedings of 34th COMPCON, IEEE March 1989

S. McGeady, "A Programmer's view of the 80960 Architectuer", Proceedings of 34th COMPCON, IEEE, March 1989

S. McGeady, "The i960CA SuperScalar Implementation of the 80960 Architecture", Proceedings of 35th COMPCON, IEEE, March 1990