MO401

Arquitetura de Computadores I

2006

Prof. Paulo Cesar Centoducatte

ducatte@ic.unicamp.br

www.ic.unicamp.br/~ducatte

MO401

Arquitetura de Computadores I

Paralelismo em Nível de Instruções
Exploração Dinâmica: Superscalar e Exemplos

"Computer Architecture: A Quantitative Approach" - (Capítulo 3)

Paralelismo em Nível de Instruções
Exploração Dinâmica

- Múltiplo Issue de Instruções por Ciclo
- Scheduling Dinâmico em Superscalar
- P6 (Pentium Pro, II, III)
- AMD Althon
- Pentium 4

Técnicas para Redução de Stalls

Capítulo 3

Capítulo 4

Technique	Reduces
Dynamic scheduling	Data hazard stalls
Dynamic branch prediction	Control stalls
Issuing multiple instructions per cycle	Ideal CPI
Speculation	Data and control stalls
Dynamic memory disambiguation	Data hazard stalls involving memory
Loop unrolling	Control hazard stalls
Basic compiler pipeline scheduling	Data hazard stalls
Compiler dependence analysis	Ideal CPI and data hazard stalls
Software pipelining and trace scheduling	Ideal CPI and data hazard stalls
Compiler speculation	Ideal CPI, data and control stalls

Múltiplos Issue de Instruções por Ciclo (CPI < 1)

- Vector Processing: Codificação explícita de iterações de loops independentes como operações em vetores
  - Instruções Multimídia têm sido adicionadas em vários processadores

- Superscalar: varia o nº.de instruções/cycle (1 a 8), escalonamento pelo compilador ou por HW (Tomasulo)
  - IBM PowerPC, Sun UltraSparc, DEC Alpha, Pentium III/4

- (Very) Long Instruction Words - (V)LIW: número fixo de instruções (4-16); escalonamento pelo compilador; operações são colocadas em templates (TBD)
  - Intel Architecture-64 (IA-64) 64-bit address
    - » Também chamado: "Explicitly Parallel Instruction Computer (EPIC)"

- Antecipação de múltiplas instruções nos leva a Instructions Per Clock Cycle (IPC) vs. CPI

CPI < 1: Issuing Múltiplas Instruções/Cycle

- Superscalar MIPS: 2 instruções, 1 FP & 1 qualquer
  - Fetch: 64-bits/ciclo clock; Int a esquerda e FP a direita
  - Somente issue a 2ª instrução se a 1ª instrução issues
  - load/store FP executado na unidade de inteiros

Type	*Pipe Stages*				
Int. instruction	IF	ID	EX	MEM	WB
FP instruction	IF	ID	EX	MEM	WB
Int. instruction		IF	ID	EX	MEM	WB
FP instruction		IF	ID	EX	MEM	WB
Int. instruction			IF	ID	EX	MEM	WB
FP instruction			IF	ID	EX	MEM	WB

- 1 ciclo load delay expande para 3 instruções em stall
  - Nem a instrução a direita e nem as instruções no próximo slot podem ser executadas

CPI < 1: Issuing Múltiplas Instruções/Cycle

- Superscalar: 2 instruções, 1 FP (3 ciclos de EX) & 1 qualquer
  - load/store FP executado na unidade de inteiros

Type	*Pipe Stages*							
Int. instruction	IF	ID	EX	MEM	WB			
FP instruction	IF	ID	EX	EX	EX	WB		
Int. instruction		IF	ID	EX	MEM	WB		
FP instruction		IF	ID	EX	EX	EX	WB	
Int. instruction			IF	ID	EX	MEM	WB	
FP instruction			IF	ID	EX	EX	EX	WB

Múltiplos Issues

- Issue Packet: grupo de instruções na unidade de fetch que podem ser despachadas em 1 mesmo ciclo de clock
  - Se a instrução causa um hazard estrutural ou hazard de dados devido a uma instrução anterior em execução ou a uma instrução anterior no issue packet, então ela não pode ser despachada
  - 0 a N instruções são despachadas por ciclo, para N-issue

Múltiplos Issues

- O Issue de múltiplas instruções (verificação de hazards) em 1 ciclo pode limitar o período do clock: $O(n^2-n)$ comparações

  - Issue Stage: Usuamente é dividido e executado em pipelined
    - » 1º estágio decide quantas instruções do packet podem ser despachadas,
    - » 2º estágio examina os hazards para as instruções selecionadas e aquelas que já foram despachadas

  - => alta penalidade em branches => precisão da predição torna-se muito importante

Desafios do Múltiplo Issue

- Quando restrita a Integer/FP é mais simples para o HW:
  - Pode conseguir CPI de 0.5 para programas com exatamente 50% de operações de FP e sem hazards
- Se mais instruções são despachadas ao mesmo tempo, maiores dificuldades para decodificar e issue:
  - para 2-scalar => examina 2 opcodes, 6 registradores & decide se 1 ou 2 instruções serão issue; (N-issue ~$O(N^2-N)$ comparações)
  - Register file: precisa de 2xN leituras and 1xN escritas/cycle
  - Lógica de Rename: deve ser capaz de renomear o mesmo registrador múltiplas vezes no mesmo ciclo!

Desafios do Múltiplo Issue

- Exemplo, considere 4-way issue:

```
add r1, r2, r3          add p11, p4, p7
sub r4, r1, r2    ⇒     sub p22, p11, p4
lw  r1, 4(r4)           lw  p23, 4(p22)
add r5, r1, r2          add p12, p23, p4
```

Imagine como fazer essa transformação em um único ciclo!

- Result Buses: Precisa suportar múltiplos complete instructions/ciclo

  » É preciso múltiplos buses com lógica de casamento associativo para cada reservation station.

  » Ou, é preciso múltiplos caminhos de forwarding

Dynamic Scheduling em Superscalar (forma simple)

- Como fazer issue de duas instruções e manter in-order instruction issue no algoritmo de Tomasulo?
  - Assuma 1 oper. inteira + 1 oper. floating point
  - 1 controle de Tomasulo para inteiro e 1 para floating point
- Issue de 2X Clock Rate, assim o issue mantem-se em ordem
- Somente loads/stores causam dependências entre issue de operações de inteiros e de FP:
  - Trocar load reservation station por uma load queue; operandos serão lidos na ordem em que são buscados
  - Load verifica endereços na Store Queue para evitar RAW
  - Store verifica endereços na Load Queue para evitar WAR e WAW

Register Renaming, Virtual Registers versus Reorder Buffers

- Uma alternativa para o Reorder Buffer é um conjunto de registradores virtuais maior e register renaming
- Virtual registers: mantém os registradores visíveis da arquitetura + registradores temporários
  - Substitui as funções do reorder buffer e das reservation stations
- O processo de Renaming mapeia os nomes dos regs da arquitetura em regs. do conjunto virtual de registradores
  - O subset dos regs. Virtuais modificados contém os regs visíveis da arquitetura
- Simplifica o instruction commit: marca o registrador como não especulativo, libera registradores com valores antigos
- Adicionado 40-80 registradores extras: Alpha, Pentium,…
  - O tamanho limita o no. de instruções em execução (usado até o commit)

Quanto (ou Quando) Se Deve Especular?

- Especulação (pró): descobre eventos que podem de alguma forma parar (stall) o pipeline (cache misses)
- Especulação (contra): especular tem custos se eventos excepcionais ocorrem quando a especulação for errada
- Solução típica: permitir especulação somente até onde eventos excepcionais não sejam (muito) caros (1st-level cache miss)
- Quando eventos excepcionais caros ocorrem (2nd-level cache miss or TLB miss) o processador espera até que a instrução que causou o evento não seja mais especulativa para tratar o evento
- Assumir um único branch por ciclo:
  - poderia-se especular através de múltiplos branches!

Limites de ILP

- Os resultados dos estudos conflitam
  - Benchmarks (vectorized Fortran FP vs. integer C programs)
  - Sofisticação do Hardware
  - Sofisticação do Compilador

- Quanto de ILP está disponível usando-se os mecanismos existentes?

- Precisamos inventar novos mecanismos em HW/SW para manter a curva de desempenho dos processadores?
  - Intel MMX, SSE (Streaming SIMD Extensions): 64 bit ints
  - Intel SSE2: 128 bit, incluíndo 2 64-bit Fl. Pt. por clock
  - Motorola AltaVec: 128 bit ints e FPs
  - Supersparc: ops Multimídia etc.

Limites para ILP

Modelo do HW inicial; compilador MIPS.

Assumir uma máquina ideal/perfeita:

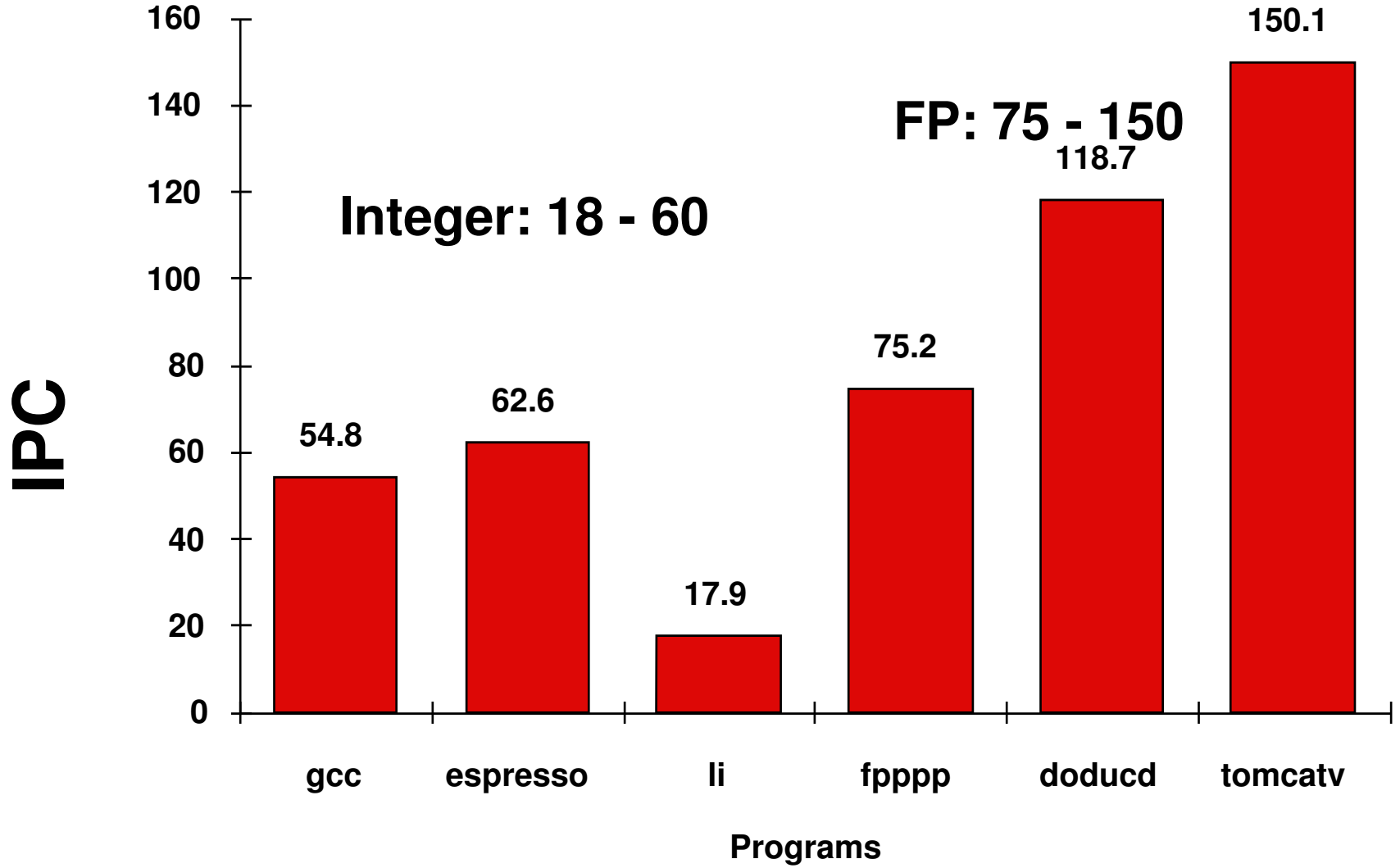
1. *Register Renaming* – infinitos regs. virtuais
   => todos os hazards WAW & WAR serão evitados
2. *Branch prediction* – perfeito; não há mispredictions
3. *Jump prediction* – todos jumps são preditos corretamente

   2 & 3 => máquina com especulação perfeita e buffer para instruções disponíveis "infinito"

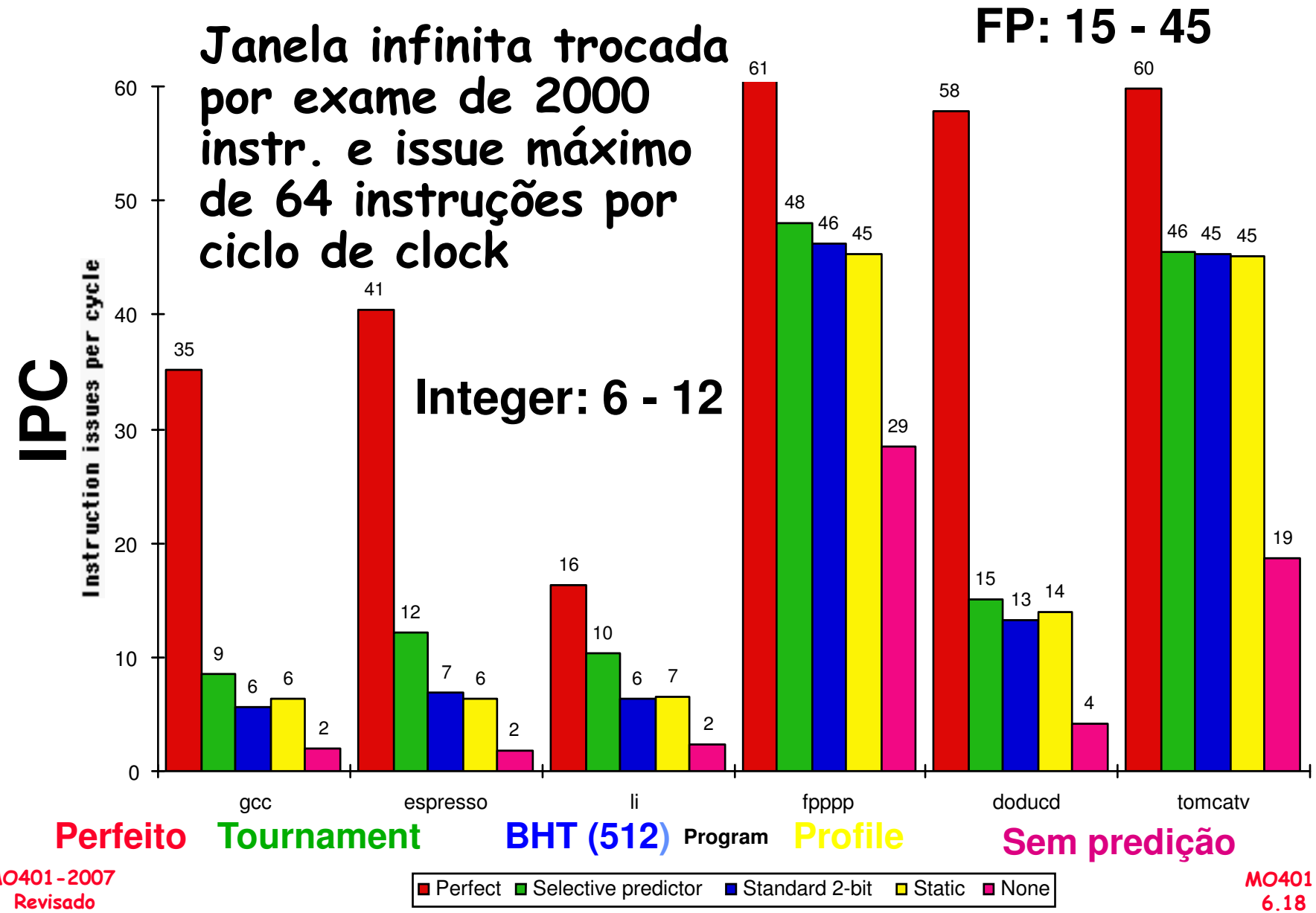
4. *Memory-address alias analysis* – endereços são conhecidos e store pode ser movido para antes de um load (endereços diferentes)

Também:
número ilimitado de instructions issued/clock cycle; cache perfeita; latência de 1 ciclo para todas as instruções (FP *,/);

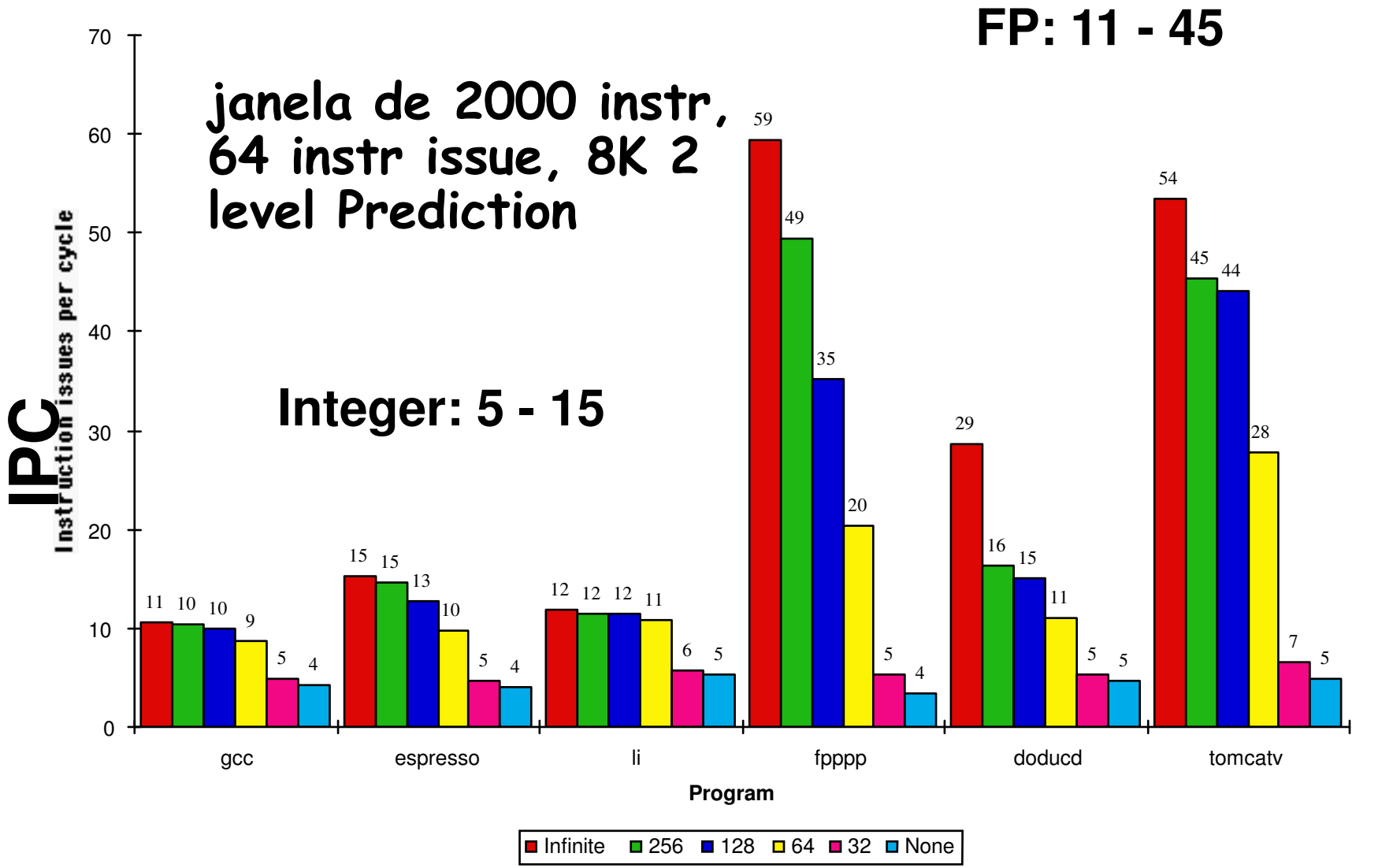
Limite Superior para ILP: Máquina Ideal



HW mais Real: Impacto dos Branches



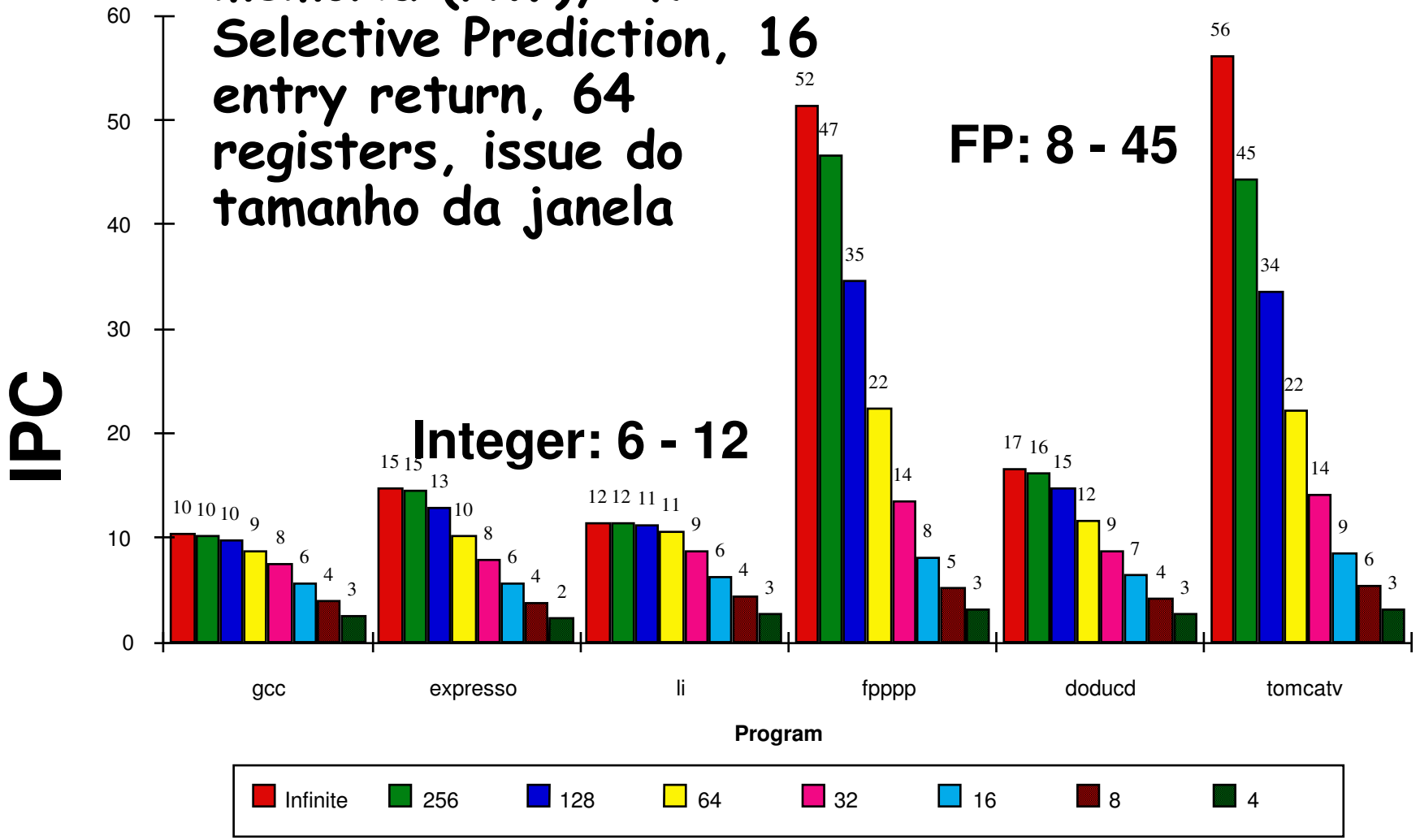
HW mais Real : Impacto de Register Renaming



Infinito 256 128 64 32 Nenhum

HW Realista: Impacto da Janela

Sem conflito de memória (HW), 1K Selective Prediction, 16 entry return, 64 registers, issue do tamanho da janela



Como Ultrapassar esses Limites para ILP?

- Hazards WAR e WAW na memória:

  elimina os hazards WAW e WAR através de register renaming, mas não no uso da memória

- Dependências desnecessárias (compilador não faz unrolling loops -> dependência na váriavel de controle de iterações)

- Sobrepor os limites do fluxo de dados: value prediction, predizer o valor e especular

- Address value prediction and speculation
  - Predição de endereços e especulação por reordenação de loads e stores

Workstation Microprocessors 3/2001

Processor	Alpha 21264B	AMD Athlon	HP PA-8600	IBM Power3-II	Intel Pentium III	Intel Pentium 4	MIPS R12000	Sun Ultra-II	Ult
Clock Rate	833MHz	1.2GHz	552MHz	450MHz	1.0GHz	1.5GHz	400MHz	480MHz	900
Cache (I/D/L2)	64K/64K	64K/64K/256K	512K/1M	32K/64K	16K/16K/256K	12K/8K/256K	32K/32K	16K/16K	32K
Issue Rate	4 issue	3 x86 instr	4 issue	4 issue	3 x86 instr	3 x ROPs	4 issue	4 issue	4 i
Pipeline Stages	7/9 stages	9/11 stages	7/9 stages	7/8 stages	12/14 stages	22/24 stages	6 stages	6/9 stages	14/15
Out of Order	80 instr	72ROPs	56 instr	32 instr	40 ROPs	126 ROPs	48 instr	None	N
Rename regs	48/41	36/36	56 total	16 int/24 fp	40 total	128 total	32/32	None	N
BHT Entries	4K ×9-bit	4K ×2-bit	2K ×2-bit	2K ×2-bit	>= 512	4K ×2-bit	2K ×2-bit	512 ×2-bit	16K
TLB Entries	128/128	280/288	120 unified	128/128	32I / 64D	128I/65D	64 unified	64I/64D	128I
Memory B/W	2.66GB/s	2.1GB/s	1.54GB/s	1.6GB/s	1.06GB/s	3.2GB/s	539 MB/s	1.9GB/s	4.8
Package	CPGA-588	PGA-462	LGA-544	SCC-1088	PGA-370	PGA-423	CPGA-527	CLGA-787	1368
IC Process	0.18µ 6M	0.18µ 6M	0.25µ 2M	0.22µ 6m	0.18µ 6M	0.18µ 6M	0.25µ 4M	0.29µ 6M	0.18
Die Size	115mm^2	117mm^2	477mm^2	163mm^2	106mm^2	217mm^2	204mm^2	126 mm^2	210
Transistors	15.4 million	37 million	130 million	23 million	24 million	42 million	7.2 million	3.8 million	29 m
Est mfg cost*	$160	$62	$330	$110	$39	$110	$125	$70	$1
Power(Max)	75W*	76W	60W*	36W*	30W	55W(TDP)	25W*	20W*	6
Availability	1Q01	4Q00	3Q00	4Q00	2Q00	4Q00	2Q00	3Q0	4(

- **Max Issue: 4 Instruções (maioria das CPUs)**
  Max Rename Registers: 128 (Pentium 4)
  Max BHT: 4K x 9 (Alpha 21264B), 16Kx2 (Ultra III)
  Max Window Size: 126 Intruções (Pent. 4)
  Max Pipeline: 22/24 Estágios (Pentium 4)

fonte: Microprocessor Report, www.MPRonline.com

SPEC 2000 Performance 3/2001 *fonte: Microprocessor Report, www.MPRonline.com*

cessor	Alpha 21264B	AMD Athlon	HP PA-8600	IBM Power 3-II	Intel PIII	Intel P4	MIPS R12000	Sun Ultra-II	Sun Ultra-III
tem or therboard	Alpha ES40 Model 6	AMD GA-7ZM	HP9000 j6000	RS/6000 44P-170	Dell Prec. 420	Intel D850GB	SGI 2200	Sun Enterprs 450	Sun Blade 100
ck Rate	833MHz	1.2GHz	552MHz	450MHz	1GHz	1.5GHz	400MHz	480MHz	900MHz
rnal Cache	8MB	None	None	8MB	None	None	8MB	8MB	8MB
.gzip	392	n/a	376	230	545	553	226	165	349
.vpr	452	n/a	421	285	354	298	384	212	383
.gcc	617	n/a	577	350	401	588	313	232	500
.mcf	441	n/a	384	498	276	473	563	356	474
.crafty	694	n/a	472	304	523	497	334	175	439
.parser	360	n/a	361	171	362	472	283	211	412
.eon	645	n/a	395	280	615	650	360	209	465
.perlbmk	526	n/a	406	215	614	703	246	247	457
.gap	365	n/a	229	256	443	708	204	171	300
.vortex	673	n/a	764	312	717	735	294	304	581
.bzip2	560	n/a	349	258	396	420	334	237	500
.twolf	658	n/a	479	414	394	403	451	243	473
Cint_base2000	518	n/a	417	286	454	524	320	225	438
.wupside	529	360	340	360	416	759	280	284	497
.swim	1,156	506	761	279	493	1,244	300	285	752
.mgrid	580	272	462	319	274	558	231	226	377
.applu	424	298	563	327	280	641	237	150	221
.mesa	713	302	300	330	541	553	289	273	469
.galgel	558	468	569	429	335	537	989	735	1,266
.art	1,540	213	419	969	410	514	995	920	990
.equake	231	236	347	560	249	739	222	149	211
.facerec	822	411	258	257	307	451	411	459	718
.ammp	488	221	376	326	294	366	373	313	421
.lucas	731	237	370	284	349	764	259	205	204
.fma3d	528	365	302	340	297	427	192	207	302
.sixtrack	340	256	286	234	170	257	199	159	273
.aspi	553	278	523	349	371	427	252	189	340
Cfp_base2000	590	304	400	356	329	549	319	274	427

1.5X 3.8X

1.2X 1.6X

1.7X

Conclusões

- 1985-2000: 1000X desempenho
  - Lei de Moore para transistores/chip => Lei de Moore para Desempenho/MPU

- Hennessy: indústria tem seguido um roadmap de idéias conhecidas em 1985 para explorar ILP e a lei de Moore nos levou a 1.55X/ano
  - Caches, Pipelining, Superscalar, Branch Prediction, Out-of-order execution, …

- Limites para ILP: faz com que o aumento de desempenho no futuro necessitará termos o paralelismo explicitado pelo programador vs. paralelismo ILP implícito explorado pelo compilador e pelo HW?
  - Ou voltaremos a taxas de 1.3X por ano?
  - Ou Menor que 1.3X devido o gap processador-memória;
  - Ou Menor que 1.3X devido a problemas de aquecimento?

Dynamic Scheduling no P6 (Pentium Pro, II, III)

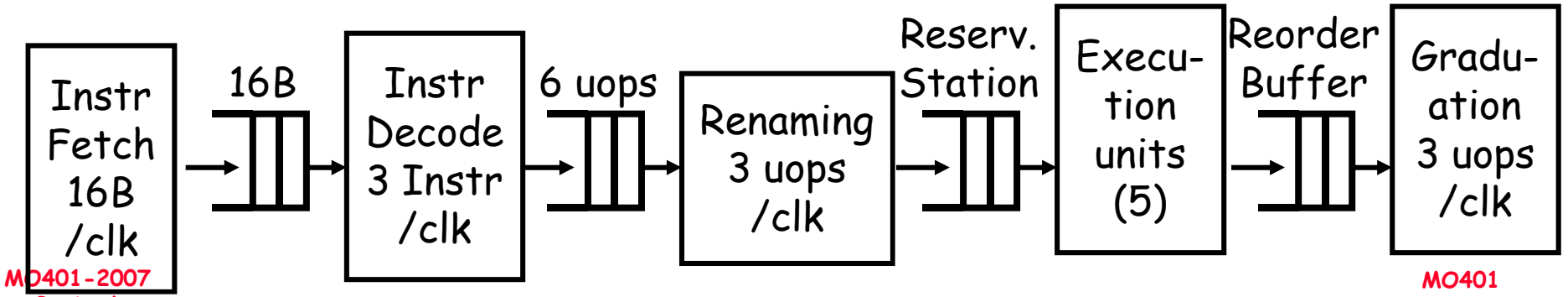
- **Q: Como Implementar Pipeline para Instruções de 1 a 17 bytes do 80x86?**
- P6 - não há pipeline para as instruções 80x86
- P6 - decode unit: instruções Intel -> micro-operações de 72-bit (~ MIPS)
- Envia as micro-operações para o reorder buffer & reservation stations
- Muitas instruções são mapeadas 1 para 4 micro-operações
- Instruções Complexas 80x86 são executadas por um microprograma convencional (8K x 72 bits) que issues longas seqüencias de micro-operações
- 14 clocks ao total no pipeline (state machines)

Dynamic Scheduling no P6

Parâmetro	80x86	micro-ops
Max. instruções issued/clock	3	6
Max. instr. completadas exec./clock		5
Max. instr. commited/clock		3
Window (Instrs no reorder buffer)		40
Número de reservations stations	20	
Número de rename registers	40	
No. integer functional units (FUs)	2	
No. floating point FUs	1	
No. SIMD Fl. Pt. FUs	1	
No. memory FUs	1 load + 1 store	

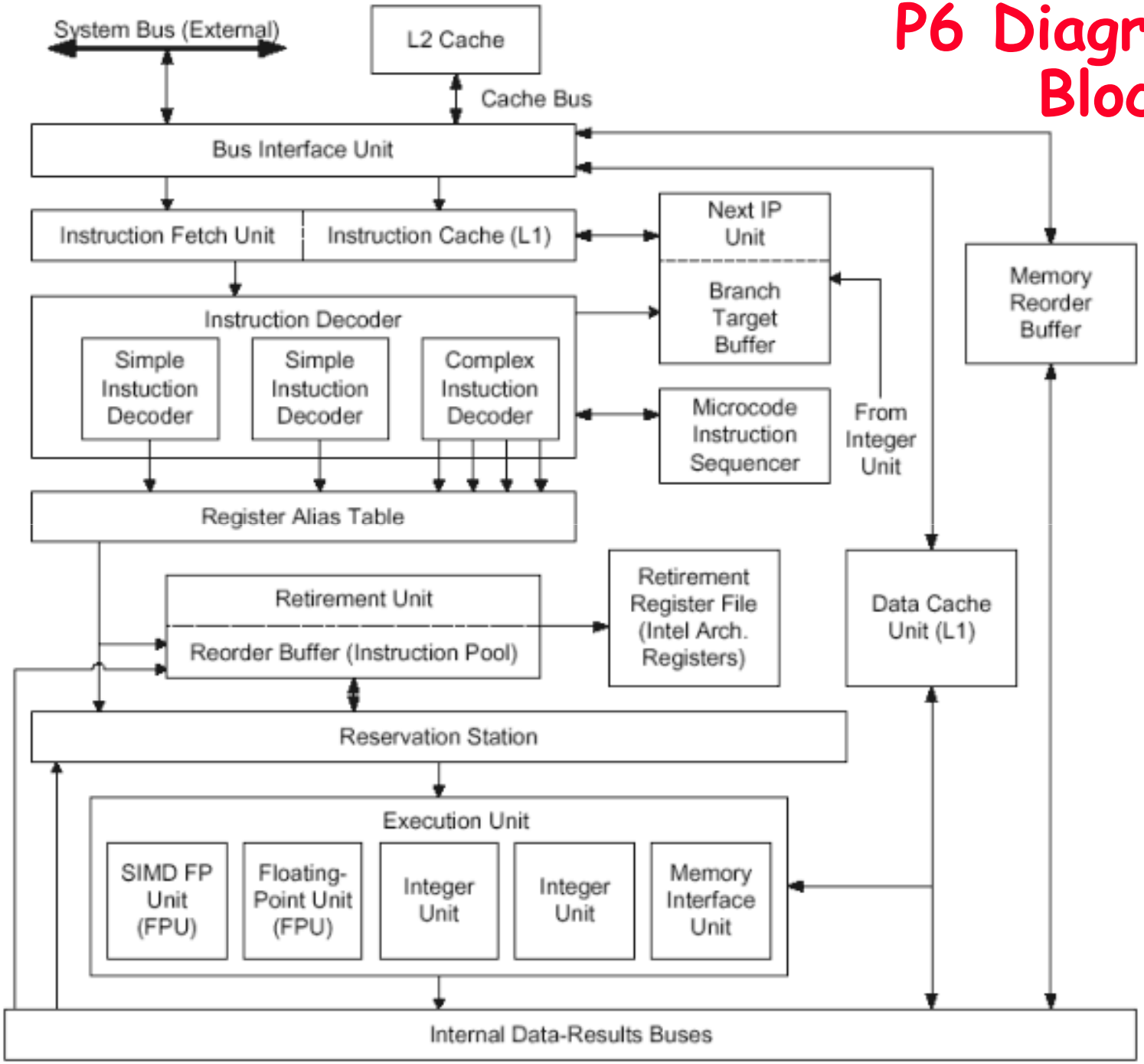
P6 Pipeline

- 14 clocks no total (~3 state machines)
- 8 estágios são usado para in-order instruction fetch, decode e issue
  - 1 ciclo de clock para determinar o tamanho da instrução 80x86 + 2 ciclos para criar a micro-operações (uops)
- 3 estágios são usados para out-of-order execution em uma das 5 unidades funcionais independentes
- 3 estágios usados para instruction commit



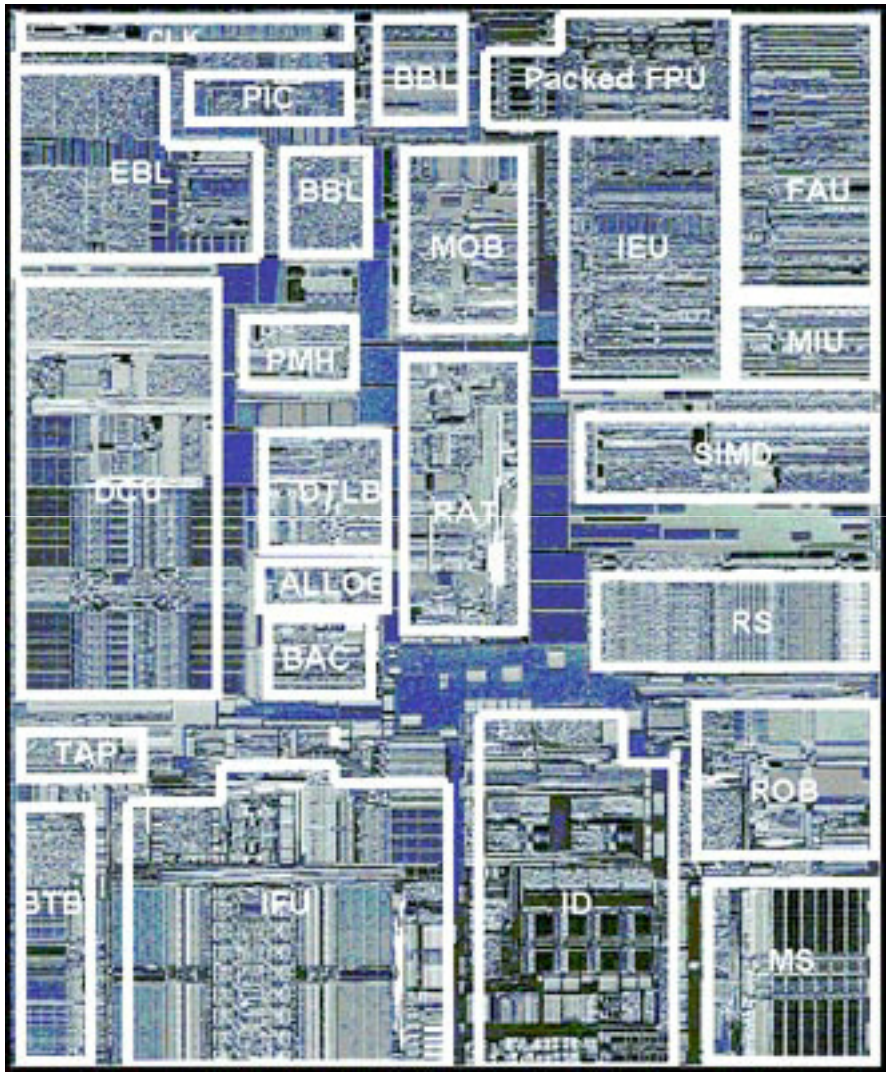
P6 Diagrama de Blocos

- IP = PC



fonte: http://www.digit-life.com/articles/pentium4/

Pentium III Die



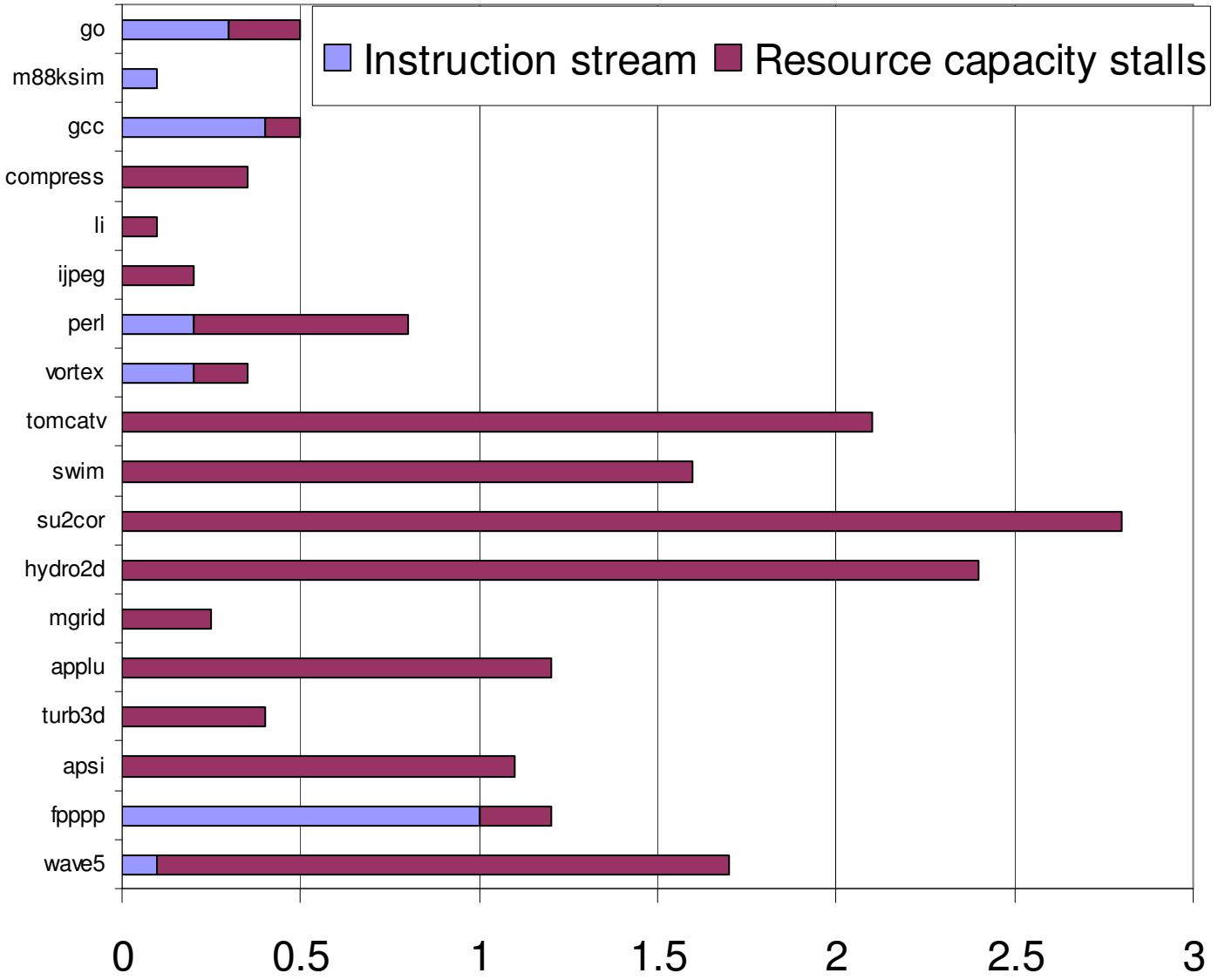
- EBL/BBL - Bus logic, Front, Back
- MOB - Memory Order Buffer
- Packed FPU - MMX Fl. Pt. (SSE)
- IEU - Integer Execution Unit
- FAU - Fl. Pt. Arithmetic Unit
- MIU - Memory Interface Unit
- DCU - Data Cache Unit

- PMH - Page Miss Handler
- DTLB - Data TLB
- BAC - Branch Address Calculator
- RAT - Register Alias Table
- SIMD - Packed Fl. Pt.
- RS - Reservation Station
- BTB - Branch Target Buffer
- IFU - Instruction Fetch Unit (+I$)

- ID - Instruction Decode
- ROB - Reorder Buffer
- MS - Micro-instruction Sequencer

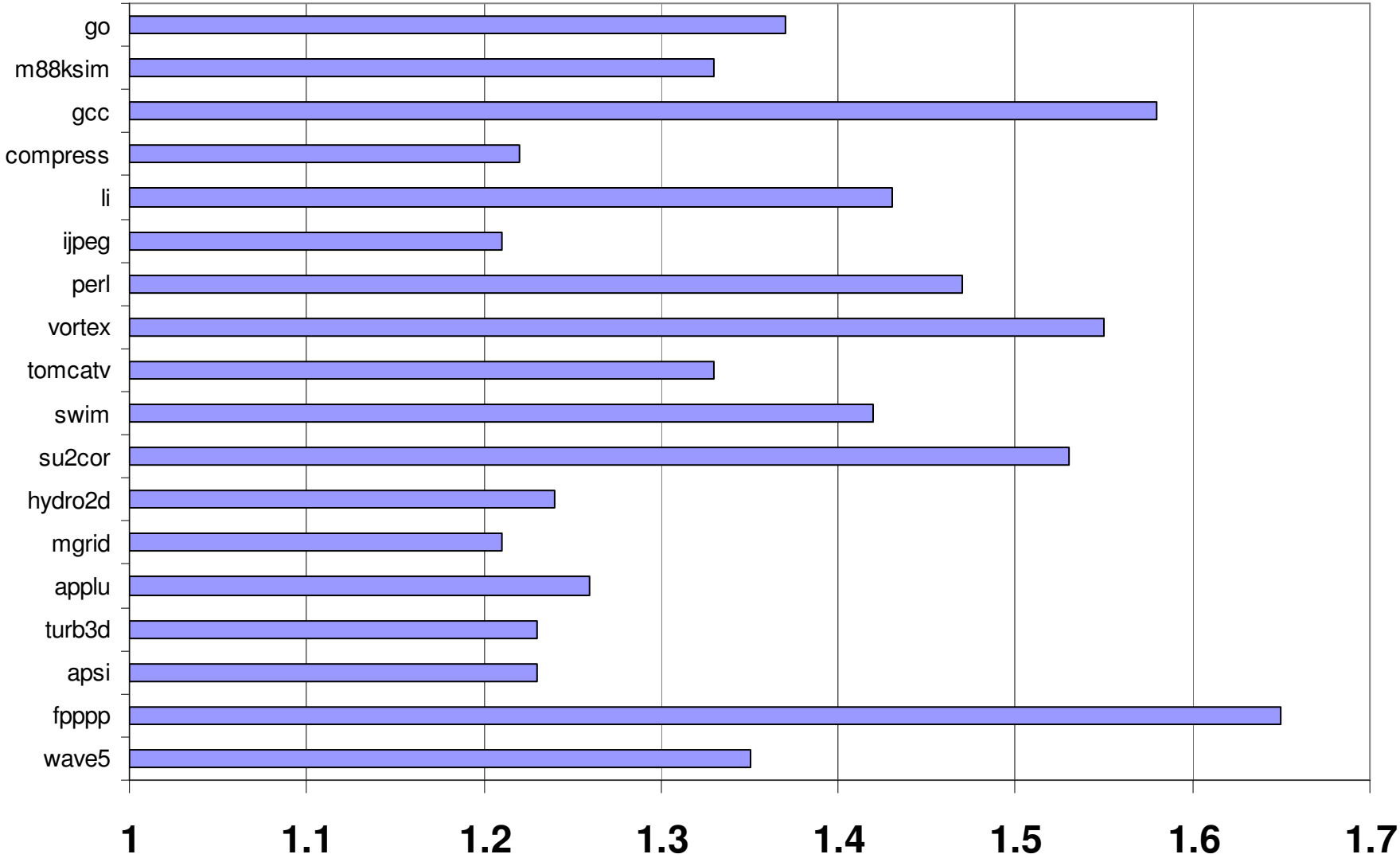
1º Pentium III, Katmai: 9.5 M transistors, 12.3 * 10.4 mm in 0.25-mi. with 5 layers of aluminum

Desempenho do P6: Stalls no decode stage
I$ misses ou falta de entradas na RS/Reorder



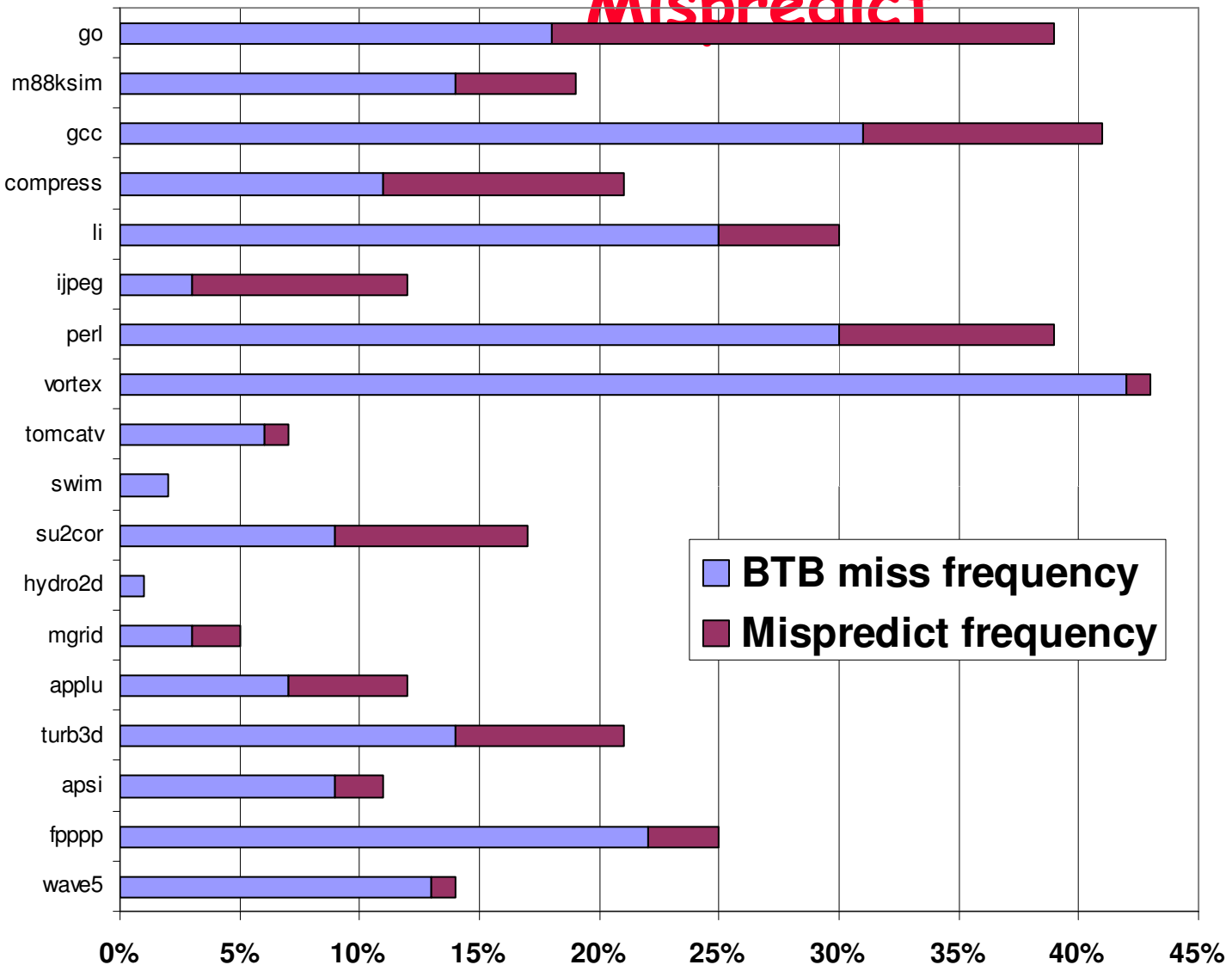
0.5 to 2.5 Stall cycles per instruction: 0.98 avg. (0.36 integer)

Desempenho P6: uops/x86 instr
200 MHz, 8KI$/8KD$/256KL2$, 66 MHz bus



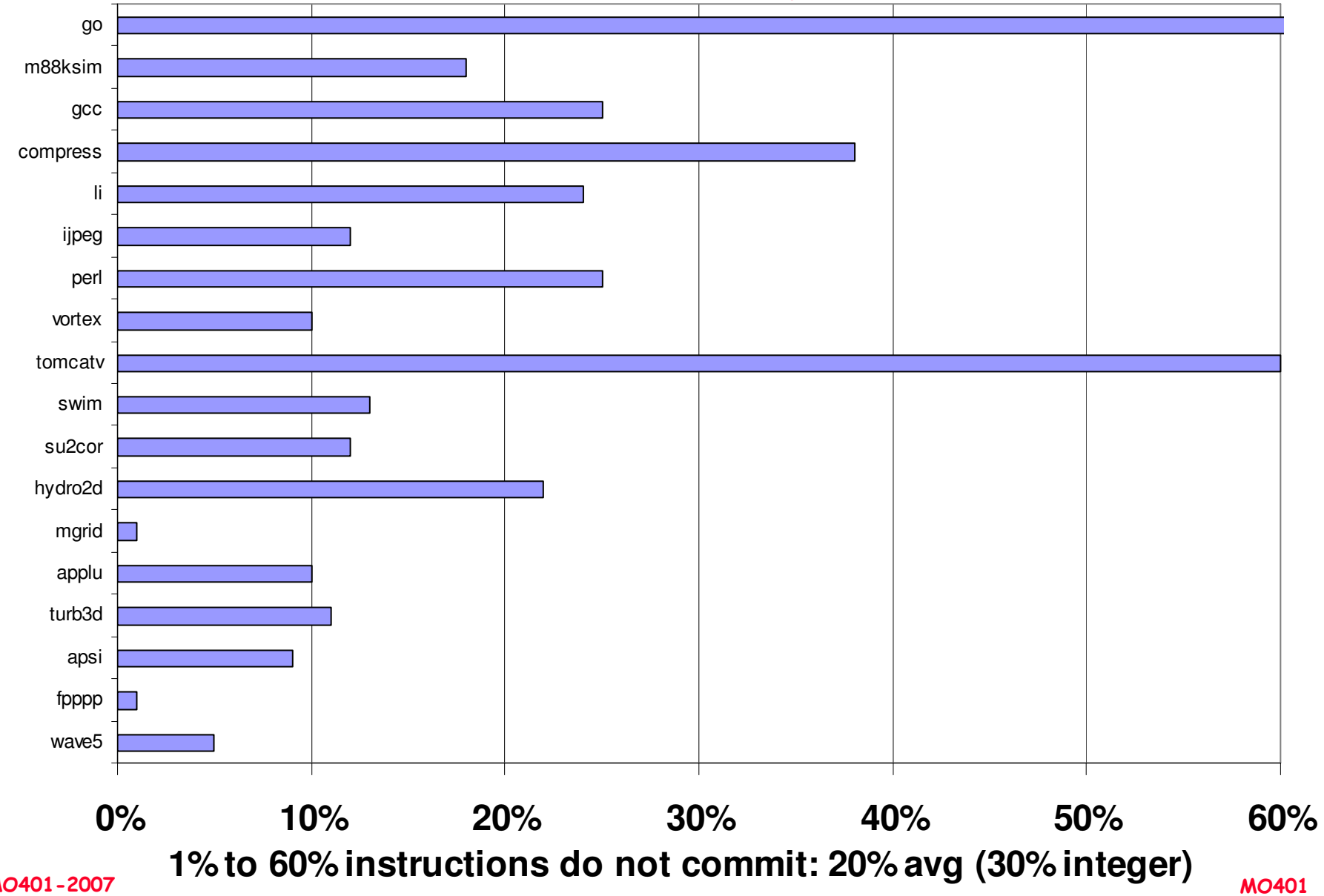
1.2 to 1.6 uops per IA-32 instruction: 1.36 avg. (1.37 integer)

Desempenho P6: taxa de Branch Mispredict

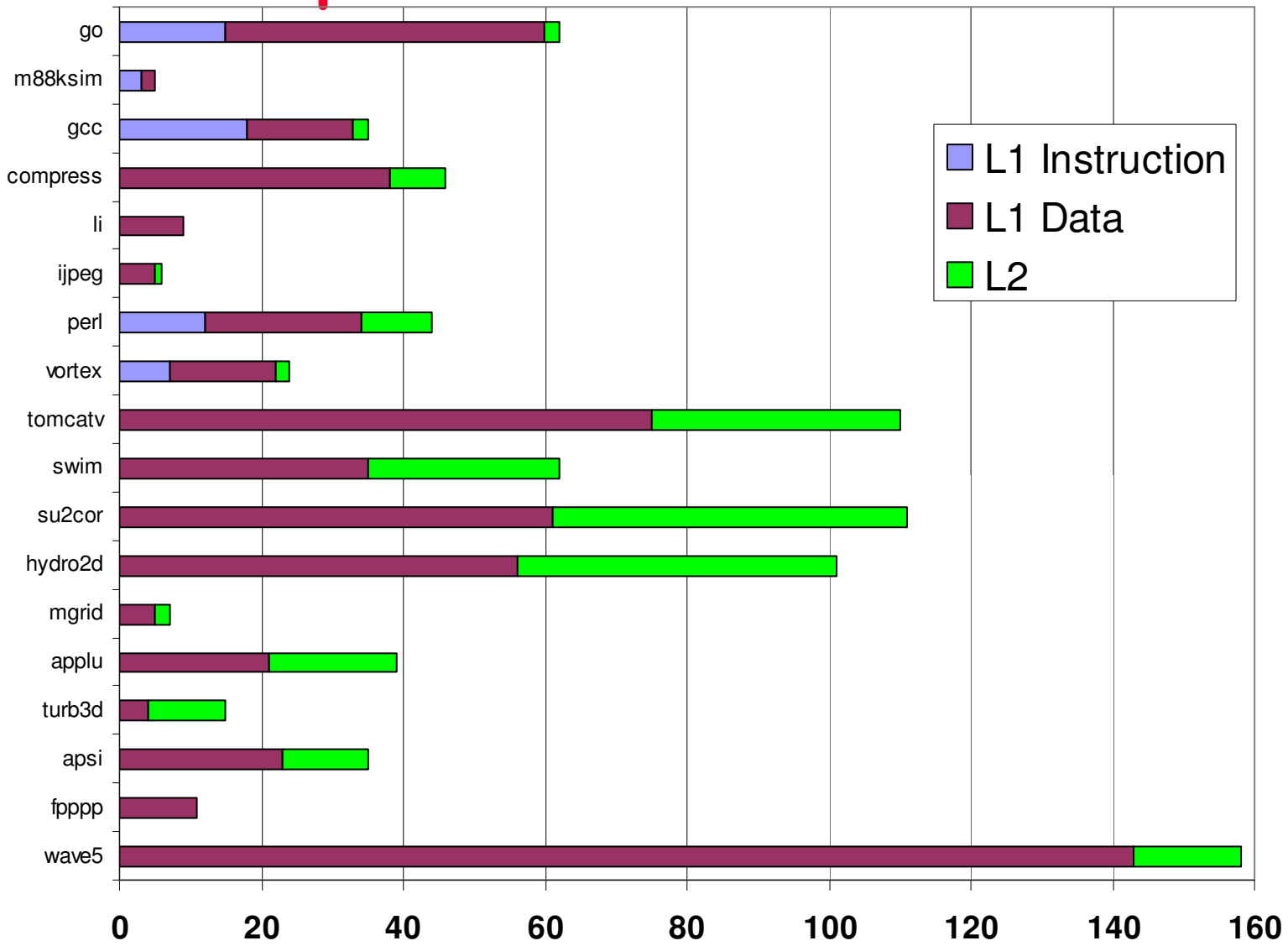


10% to 40% Miss/Mispredict ratio: 20% avg. (29% integer)

Desempenho P6: Speculação (% instruções issued que não commit)

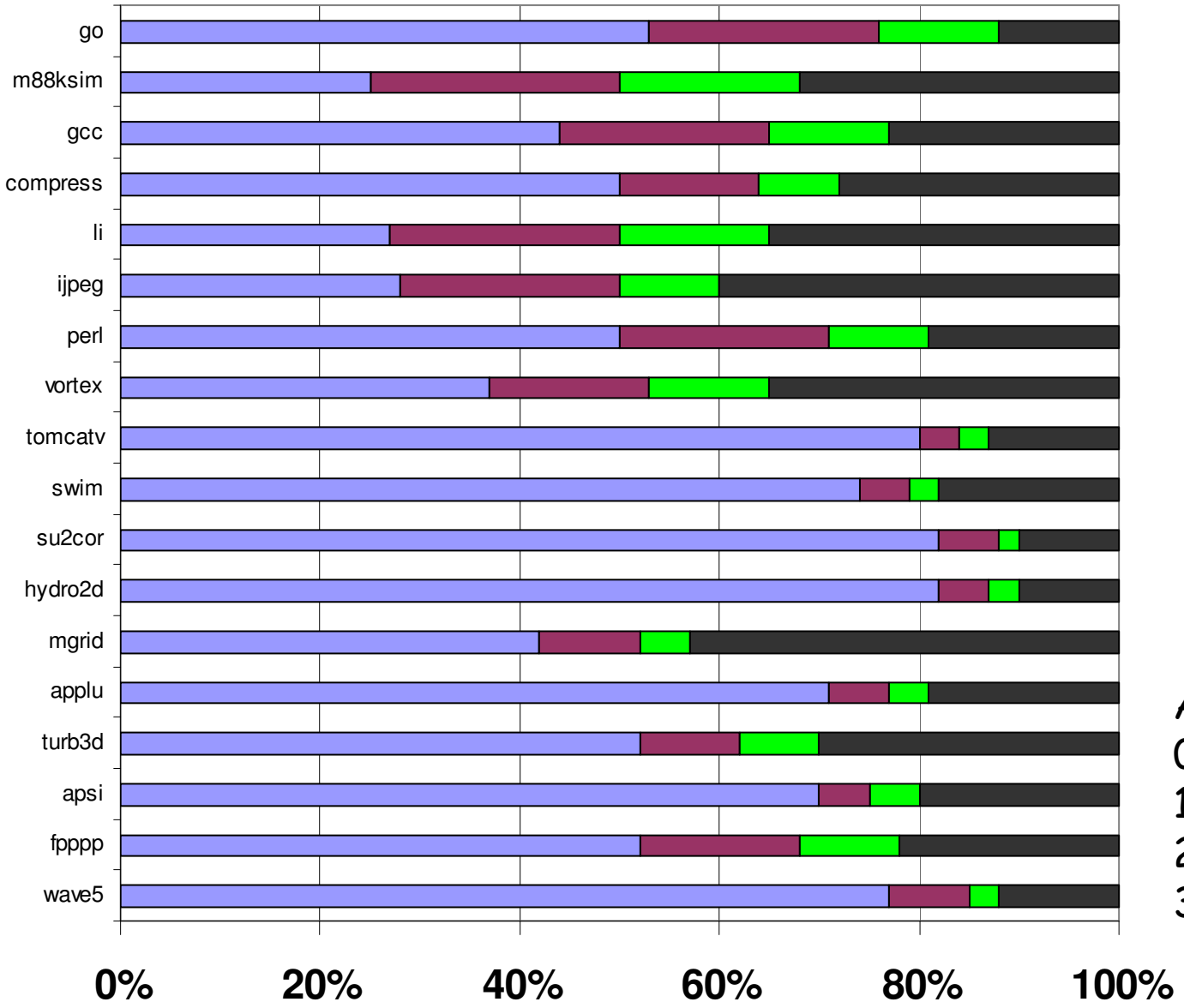


Desempenho P6: Cache Misses/1k instr



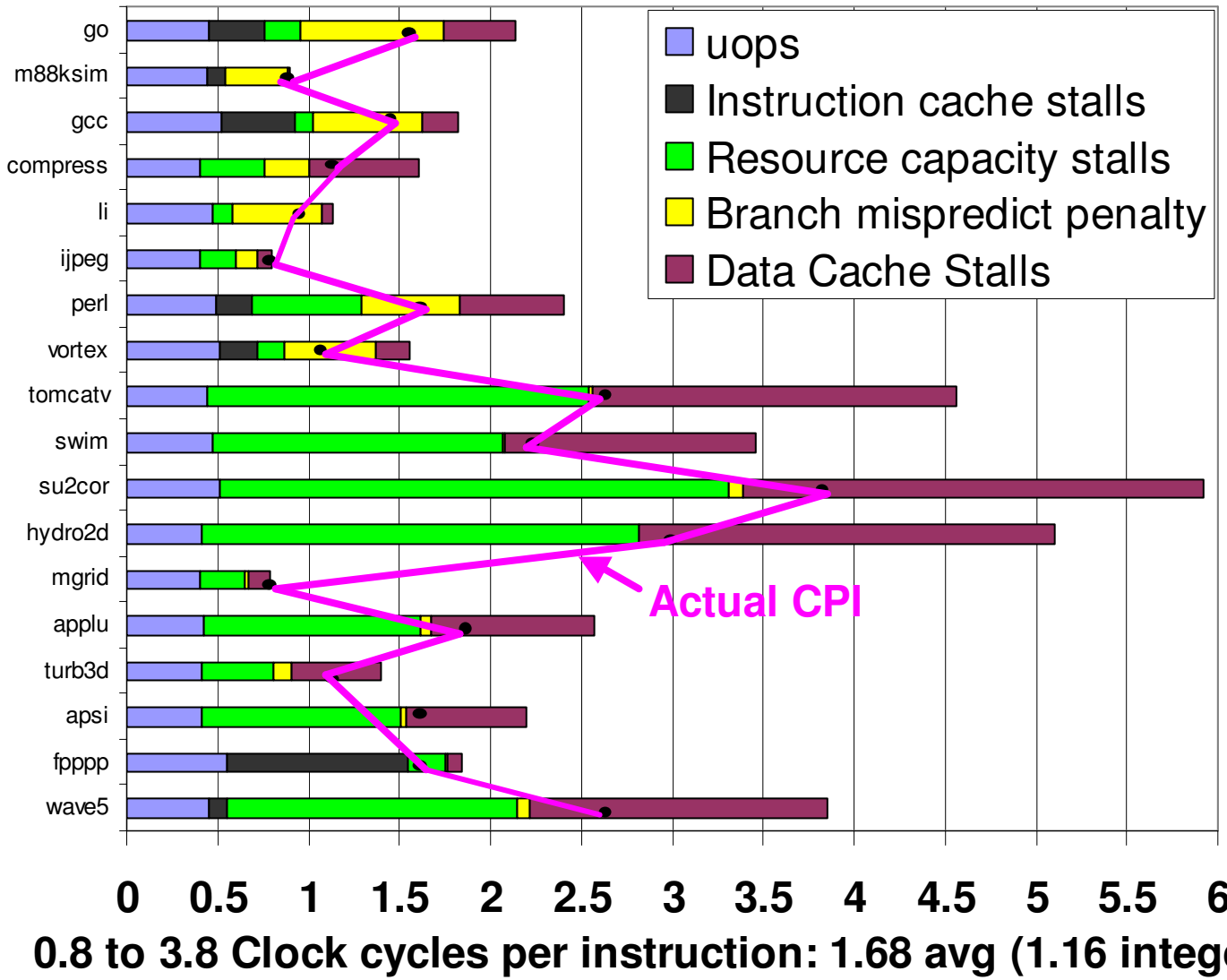
10 to 160 Misses per Thousand Instructions: 49 avg (30 integer)

Desempenho P6: uops commit/clock



Average	Integer
0: 55%	0: 40%
1: 13%	1: 21%
2: 8%	2: 12%
3: 23%	3: 27%

Beneficios Dinâmicos do P6?
Soma das partes CPI vs. CPI atual



Ratio of sum of parts vs. actual CPI: 1.38X avg. (1.29X integer)

AMD Althon

- Similar à micro-arquitetura do P6 (Pentium III), porém com mais recursos
- Transistores: PIII 24M v. Althon 37M
- Die Size: 106 mm^2 v. 117 mm^2
- Power: 30W v. 76W
- Cache: 16K/16K/256K v. 64K/64K/256K
- Window size: 40 vs. 72 uops
- Rename registers: 40 v. 36 int + 36 Fl. Pt.
- BTB: 512 x 2 v. 4096 x 2
- Pipeline: 10-12 estágios v. 9-11 stages
- Clock rate: 1.0 GHz v. 1.2 GHz
- Memory bandwidth: 1.06 GB/s v. 2.12 GB/s

Pentium 4

- Ainda usa mapeamento de 80x86 para micro-ops
- P4 tem branch predictor melhor, mais FUs
- Instruction Cache mantém micro-operações vs. 80x86 instruções
  - Não há decodificação de 80x86 em cache hit
  - denominado "trace cache" (TC)
- Memory bus mais rápido: 400 MHz v. 133 MHz
- Caches
  - Pentium III: L1I 16KB, L1D 16KB, L2 256 KB
  - Pentium 4: L1I 12K uops, L1D 8 KB, L2 256 KB
  - Block size: PIII 32B v. P4 128B; 128 v. 256 bits/clock
- Clock rates:
  - Pentium III 1 GHz v. Pentium IV 1.5 GHz
  - Pipeline de 14 estágios vs. pipeline de 24 estágios

Características do Pentium 4

- Instruções Multimídia de 128 bits vs. 64 bits
  => 144 novass instruções
  - Quando são usadas pelos programas??
  - Floating Point mais rápido: executa 2 instr. 64-bit Fl. Pt. Por clock
  - Memory FU: 1 128-bit load, 1 128-store /clock para regs MMX
- Usa RAMBUS DRAM
  - Bandwidth melhor, a mesma latência de SDRAM
  - Custo 2X-3X vs. SDRAM
- ALUs: opera a 2X o clock rate para muuitas ops
- Pipeline não stall neste clock rate: uops replay
- Rename registers: 40 vs. 128
- Window: 40 v. 126
- BTB: 512 vs. 4096 entradas

Pipeline: Pentium, Pentium Pro, Pentium 4

P5 Microarchitecture

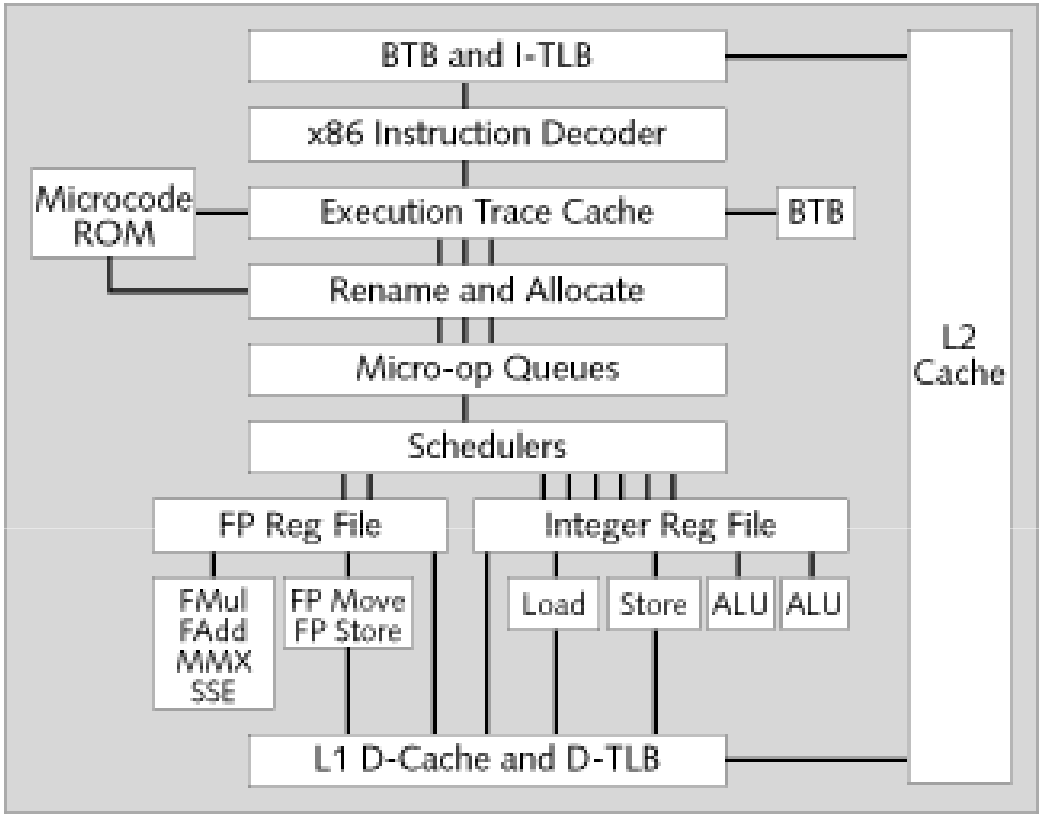
P6 Microarchitecture

NetBurst Microarchitecture

- **Pentium (P5) = 5 estágios**
  Pentium Pro, II, III (P6) = 10 estágios (1 ciclo ex)
  Pentium 4 (NetBurst) = 20 estágios (sem decode)

fonte "Pentium 4 (Partially) Previewed," Microprocessor Report, 8/28/00

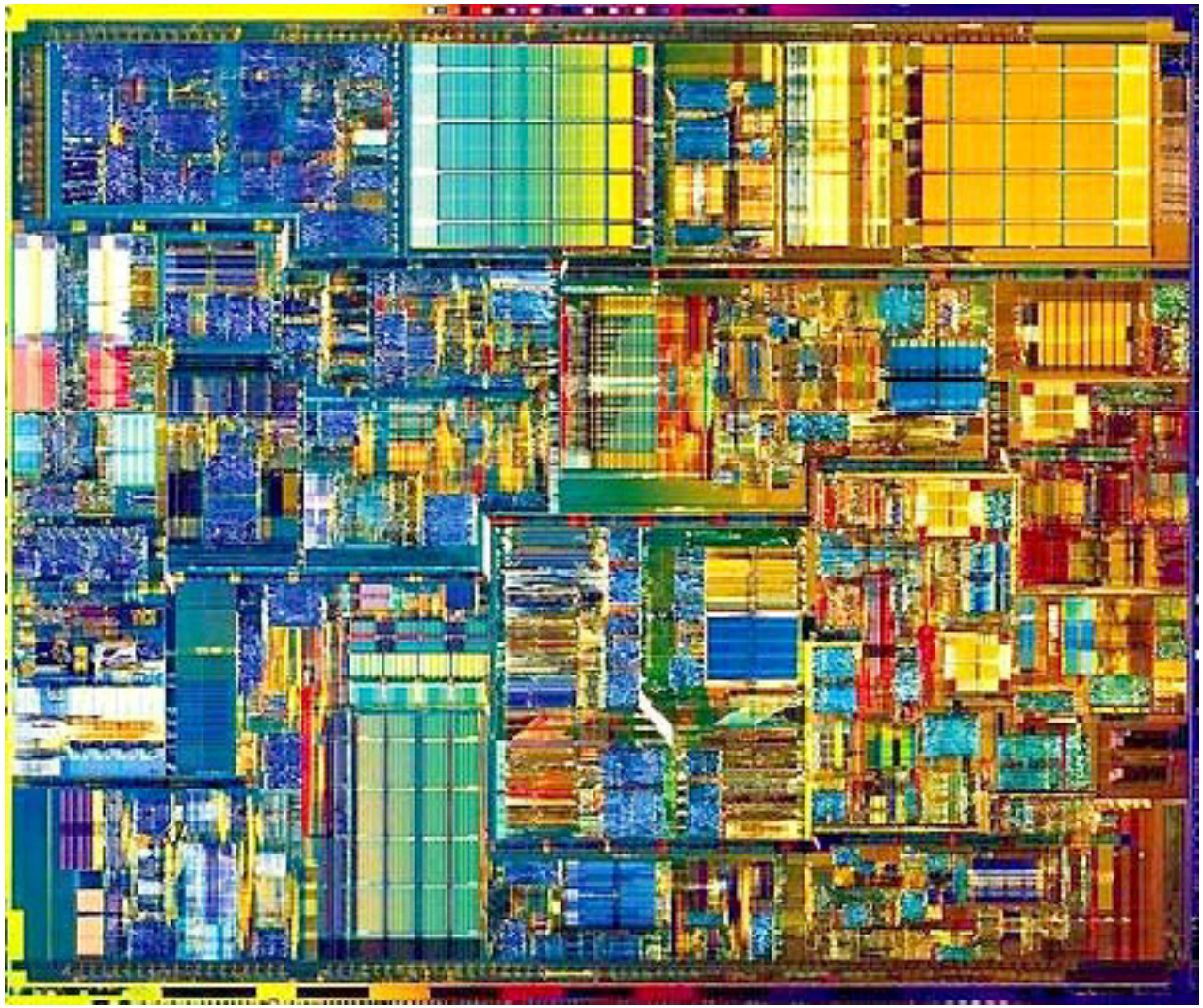
Diagrama de Bloco do Pentium 4 Micro-arquitetura



- BTB = Branch Target Buffer (branch predictor)
- I-TLB = Instruction TLB, Trace Cache = Instruction cache
- RF = Register File; AGU = Address Generation Unit
- "Double pumped ALU" means ALU clock rate 2X => 2X ALU F.U.s

From "Pentium 4 (Partially) Previewed," Microprocessor Report, 8/28/00

Pentium 4 Die



- 42M Xtors
  - PIII: 26M
- 217 mm^2
  - PIII: 106 mm^2
- L1 Execution Cache
  - Buffer 12,000 Micro-Ops
- 8KB data cache
- 256KB L2$

Benchmarks: Pentium 4 v. PIII v. Althon

- SPECbase2000
  - Int, P4@1.5 GHz: 524, PIII@1GHz: 454, AMD Althon@1.2Ghz:?
  - FP, P4@1.5 GHz: 549, PIII@1GHz: 329, AMD Althon@1.2Ghz:304
- WorldBench 2000 benchmark (business) PC World magazine, Nov. 20, 2000 (bigger is better)
  - P4 : 164, PIII : 167, AMD Althon: 180
- Quake 3 Arena: P4 172, Althon 151
- SYSmark 2000 composite: P4 209, Althon 221
- Office productivity: P4 197, Althon 209
- S.F. Chronicle 11/20/00: "… the challenge for AMD now will be to argue that frequency is not the most important thing-- precisely the position Intel has argued while its Pentium III lagged behind the Athlon in clock speed."

Por que?

- Instruction count é o mesmo para x86
- Clock rates: P4 > Althon > PIII
- Como o P4 pode ser mais lento?
- Time =
  Instruction count x CPI x 1/Clock rate
- Average Clocks Per Instruction (CPI) do P4 deve ser pior do que o do Althon e do PIII
- O CPI será sempre < 1.0 para programas reais?