

Avaliação de arquitetura baseada no Intel i7-930 com variações no compartilhamento e tamanho da cache L2

Daniel Gunna

Ciência da Computação

Pontifícia Universidade Católica de Minas Gerais

Belo Horizonte, MG

danielgunna1408@gmail.com

Felipe Coelho Silva

Ciência da Computação

Pontifícia Universidade Católica de Minas Gerais

Belo Horizonte, MG

felipe.c.s@ieee.org

Leonardo Palis

Ciência da Computação

Pontifícia Universidade Católica de Minas Gerais

Belo Horizonte, MG

leonardoafpalis@gmail.com

Marcelo R. dos Santos Júnior

Ciência da Computação

Pontifícia Universidade Católica de Minas Gerais

Belo Horizonte, MG

marrsantosjunior@gmail.com

Paulo Victor de Oliveira Leal

Ciência da Computação

Pontifícia Universidade Católica de Minas Gerais

Belo Horizonte, MG

pauloolileal@gmail.com

Abstract—Neste trabalho, apresentamos diferentes simulações realizadas com o SESC (Superscalar simulator), com arquiteturas baseadas no processador i7-930. Nas simulações, foram simulados os principais aspectos da arquitetura do processador. Também simulou-se variadas configurações de tamanho e compartilhamento de cache L2 a além da quantidade de núcleos envolvidas no processamento, a fim de comparar o desempenho da arquitetura original com as diferentes variações realizadas.

Index Terms—Avaliação; Análise; Intel i7-930; Cache L2; SESC;

I. INTRODUÇÃO

É de conhecimento geral que os processadores Intel Core i7 estão atualmente muito presentes no mercado. Esse processador, que contém excelentes características, é estudado nesse artigo com base em simulações por meio do SESC (Superscalar Simulator)[?], no qual é proposto melhorias e é avaliado quais as melhores condições de alterações na arquitetura.

Uma imagem que representa a arquitetura do i7 é mostrada na figura 1

A. Processador Analisado

Processador Intel® Core™ i7-930 que contém 4 núcleos com 8 threads no total tem frequência de 2.8 GHz e utiliza a tecnologia de Intel Hyper-Threading. As memórias cache do processador são: L2 de 4x256KB e L3 de 8MB.

Além disso, é o funcionamento das memórias cache podem ser observadas através da figura 2.

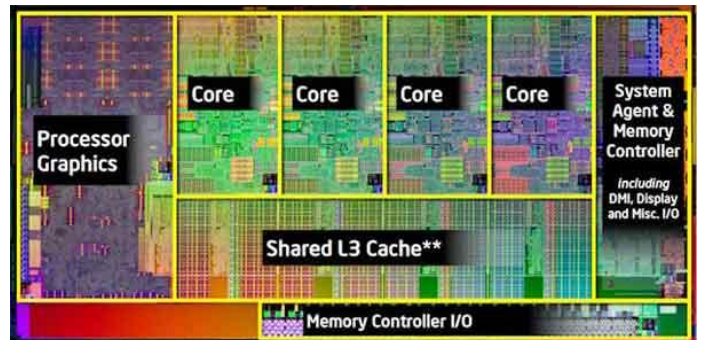


Fig. 1. Arquitetura tomada como base, dos processadores Intel i7

B. Processador Simulado

As configurações do i7 não podem ser completamente simuladas no SESC, por isso apenas as configurações primordiais do processador serão simuladas. As características simuladas do i7 serão:

- Frequência;
- Cache L1 64KB para dados e instruções;
- Cache L2 256MKB para cada núcleo individual;
- Cache L3 8MB compartilhado;
- Tamanho do chip;



Fig. 2. Arquitetura tomada como base, dos processadores Intel i7

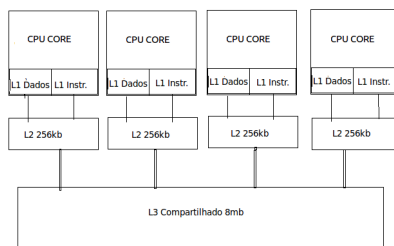


Fig. 3. Configuração Original do i7.

II. METODOLOGIA

Foi utilizado o simulador Sesc para fazer a simulação do processador especificado no item 1 seção B.

Os testes foram conduzidos variando o tamanho da Cache L2 do processador e a quantidade de núcleos envolvidas no processamento.

Utilizou-se a carga de trabalho Ocean, do benchmark SPLASH, para realizar os testes. Na Figura 1 é mostrada a configuração original do i7. Basicamente possui a organização de memória é composta por uma cache L1 de 64 Kb, com 32 Kb para dados e 32 Kb para instruções. Possui uma cache L2 para cada núcleo de 256Kb e uma cache L3 de 8Mb compartilhada entre todos os núcleos. Neste trabalho serão avaliadas duas configurações de organização de cache L2. Na primeira delas alteraremos o tamanho da cache L2 individual e removeremos a cache L3 em ambas as simulações e manteremos a cache L2 única para cada núcleo como mostrado na Figura 2. Neste caso visou-se aumentar a cache para se obter melhor localidade especial e localidade temporal, pois com uma cache L2 individual maior mais dados estariam disponíveis a cada busca de dados e os tempos de acesso seriam menores. Na outra configuração de organização apresentada a cache L2 teve seu

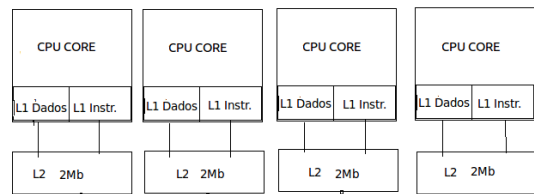


Fig. 4. Configuração Simulada 2Mb L2 para cada núcleo do i7.

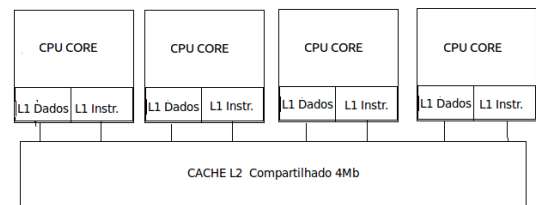


Fig. 5. Configuração Simulada 4Mb L2 compartilhada.

tamanho aumentado novamente, desta vez para 4Mb, porém ele seria compartilhada entre o 4 núcleos como mostrado na Figura 3. Objetiva-se nesta configuração observar com o compartilhamento de uma cache L2 maior interferiria na performance e nos indicadores avaliados. Em ambas as simulações manteve-se a configuração de cache L1 original do i7. Em ambas também os seguintes parâmetros significativos foram utilizados:

- Associaatividade de 8 vias;
- Política de reescrita LRU;
- Política de escrita WT;
- Protocolo MESI;
- Despacho fora de ordem.

III. RESULTADOS OBTIDOS

Apresentamos os resultados considerados mais relevantes para a proposta deste trabalho na figuras 4 e 5.

Configuração original i7 - L2 256Kb por Núcleo

Nº de núcleos	1	2	4
Miss Rate L1(Dados e intr.)	12.96% e 0.05%	12.86% e 0.06%	12.59% e 0.09%
Simulation time	769.20 msec	377.694 msec	157.890 msec
Execution time	1381.040 secs	1428.020 secs	1488.370 secs
Cache Miss L2	14458878	6973605	10431123
Média Ciclos Nucleo	2154377238	1057496331,5	442022062
IPC Médio	0.22	0.23	0.28
IPC Médio	95.40% e 78.09%	47.70 % e 39.12%	19.50% e 23.76%

Fig. 6. Tabela Simulação i7.

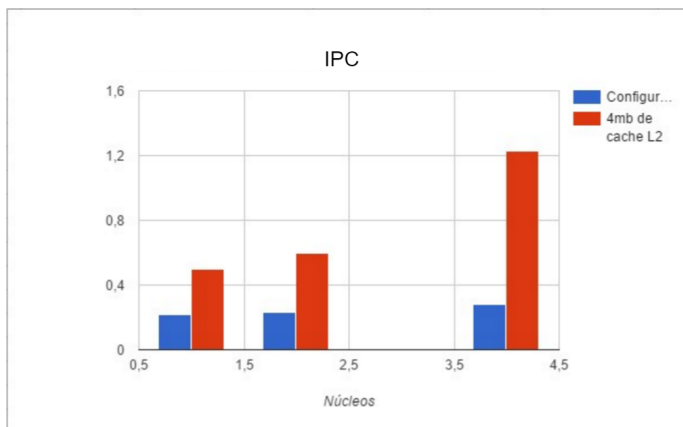


Fig. 7. Resultado da comparação entre os IPCs (Instruções por Ciclo)

É possível notar que o aumento da quantidade de núcleos na arquitetura influenciou positivamente o tempo de execução.

Além disso, o aumento da cache também impactou positivamente os resultados dos testes realizando com a carga de trabalho Ocean do SPLASH, como pode ser observado na tabela 4.

IV. CONCLUSÃO E CONSIDERAÇÕES FINAIS

Através das simulações realizadas obtemos resultados que nos levaram a conclusão que houve uma melhoria utilizando-se a configuração de cache L2 compartilhado com 4Mb, nas três quantidades de núcleos utilizadas, se comparada com a organização de cache original simulada. O que constatou-se é que com uma cache maior compartilhada a quantidade de miss de escrita e leitura da cache L2 foram menores do que na arquitetura original, o que fez com que a quantidade de ciclos por instrução subisse bastante se comparado a configuração original do i7.

É importante levar em consideração que foi apenas avaliado o impacto das variações de cache L2 neste trabalho. Por ser uma cache de segundo nível, é esperado que o seu impacto na arquitetura não seja tão grande quanto seria em uma cache

L1, por exemplo. Dessa forma, a variação da cache L1 é considerada como trabalho futuro.

Outros fatores que podem ser levados em consideração, são:

- Variação da Cache L3
- Variação abrupta de núcleos (Many-core)
- Variação da carga de trabalho (Além do Ocean)

Além disso, neste trabalho realizamos uma análise da performance das arquiteturas. É importante levar em consideração outras análises, como uma análise financeira (custo das variações), análise energética, entre outras.

V. REFERÊNCIAS

- 1 ACEVEDO, M F A probabilistic study of two-level storage hierarchies M S Th, U of Texas, Austin, Tex, Dec 1972.
- 2 AVEN, O I, ET AL Some results on distribution-free analysis of paging algorithms IEEE Trans Computrs C-25, 7 (July 1976), 737-745
- 3 BASKETT, F, AND RAFFI, A The A0 reversion model of program paging behavior Tech Rep STAN-CS- 76-579, Dept Comptr Sci, Stanford U, Stanford, Calif. Oct 1976
- 4 BELADY, L A A study of replacement algorithms for virtual storage computers IBM Syst J..5. 2 (1960), 78-101
- 5 BELL, J, CASASANT, D, AND BELL, C G An investigation of alternative cache organizations IEEE Trans Computrs. C-23, 4 (April 1974), 346-35 !
- 6 BURVILLE, P J, AND KINGMAN, J F C On a model for storage and search J Appl Probability 10 (1973), 697-70 !
- 7 COFFMAN, E G, AND DENNING, P J Operating System Theory Prentice-Hall, Englewood Cliffs, N J, 1973 Performance Analysis of Cache Memories 3 9 5
- 8 COFFMAN, E G, AND RVAN, T A A study of storage partitioning using a mathematical model of locality Comm ACM 15, 3 (March 1972), 185-190
- 9 CONTI, C J Concepts for buffer storage Comptr Group News 2, 8 (March 1969), 9-13
- 10 FERRARI, D Improving locality by critical working sets Comm A CM 17, 11 (Nov 1974), 614-620
- 11 FERRARI, D Improving program locality by strategy-oriented restructuring Information Processing, NorthHolland Pub Co, Amsterdam, 1974, pp 266-270
- 12 FRANASZEK, P A, AND WAGNER, T J Some distribution-free aspects of paging algorithm performance. J ACM21, 1 (Jan 1974), 31-39
- 13 GELENBE, E A unified approach to the evaluation of a class of replacement algorithms IEEE Trans Computrs C-22, 6 (June 1973), 611-618
- 14 HATFIELD, D J, AND GERALD, J Program restructuring for virtual memory IBM Syst J 10, 3 (1971), 168-192
- 15 HILLIER, F S, AND LIEBERMAN, G J Introduction to Operations Research Holden-Day, San Francisco, 1972
- 16 JOHNSON, J W Program restructuring for virtual memory systems MAC TR-148. M I T, Cambridge, Mass, March 1975
- 17 KING, W F Analysis of paging algorithms Proc IFIP Congress, Ljubljana, Yugoslavia, Aug 1971, 485-490

18 LENFANT, J The delay network model of program behaviour In Computer Architecture and Networks, E Gelenbe and R Muhl, Eds, North-Holland Pub Co, Amsterdam, 1974, pp 299-329

19 MADNICK, S E Storage hierarchy systems MAC TR-107, M I T, Cambridge, Mass. 1972

20 MADNICK, S E, AND DONOVAN, J J Operating Systems McGraw-Hill, New York, 1974

21 RAO, G S Performance analysis of cache memories Tech Rep No 110, SEL 76-019, Digital Systems Lab, Depts EE and Computr Sct, Stanford U., Stanford, Calif, 1975

22 RAO, B R The stack working set A characterization of spatial locality Tech Rep. No 95, Digital Systems Lab, Stanford U, Stanford, Calif, July 1975

23 RAO, B R, AND ROSSMANN, G E Cache based computer systems A tutorial Tech Rep, Palyn Assocs, San Jose, Calif, March 1974

24 RIVEST, R L On self-organizing sequential search heuristics Res Rep 61, IRIA Laborla, Le Chesnay, France, March 1974