Aluno: Igor Pereira Dourado 19204004

Disciplina: Org e Arq 2

Trabalho Exploratório 1

Meu processador

Introdução:

Detalhes do meu processador:

Nome: Intel Core i5-7200U CPU

Modelo: i5-7200U baseado em x64

Fabricante: Intel

Ano de produção: introduzido em 2016

Tecnologia de semicondutor usada: 14 nm

Número de cores: 2

Número de threads: 4

Memória cache: 4 MB

Frequência baseada em processador: 2.50 GHz

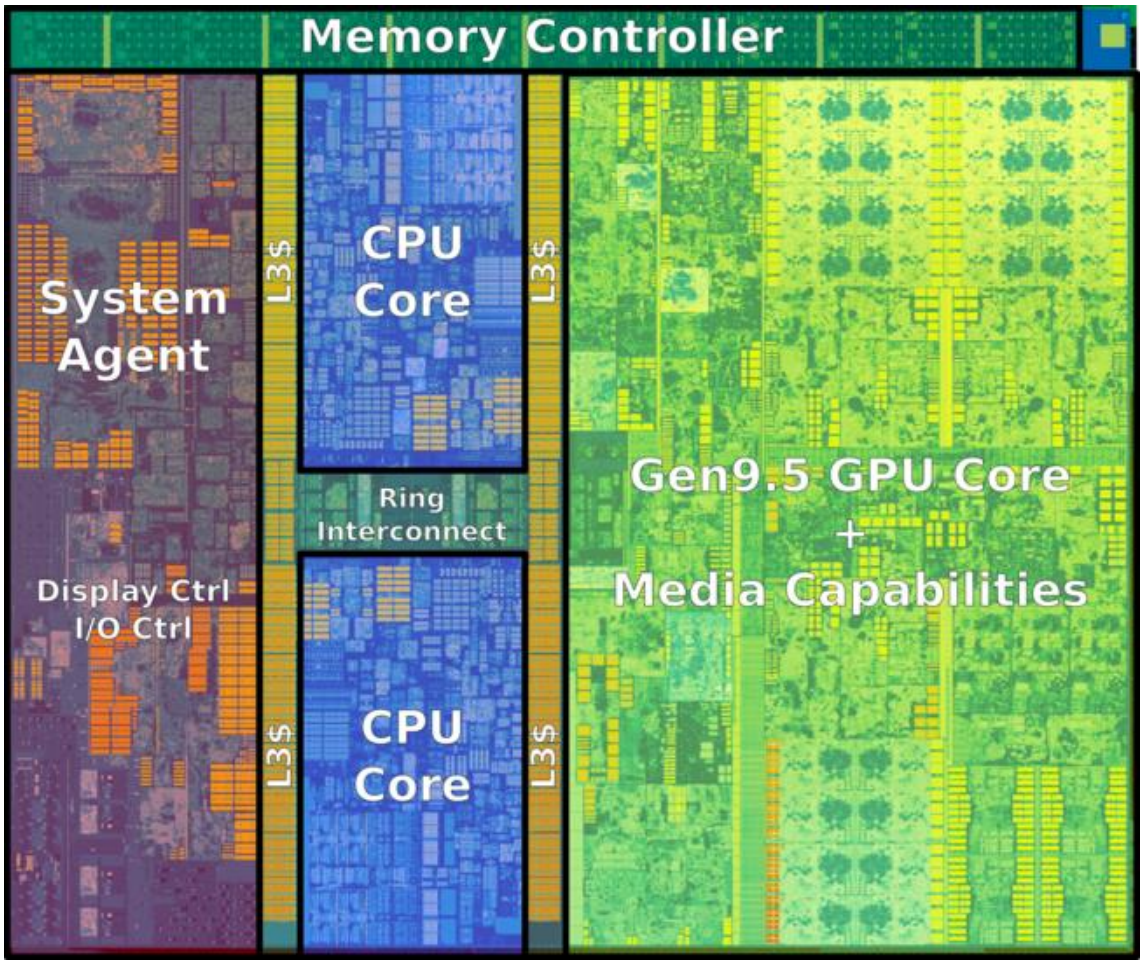
Frequência de TDP Configurável – alto: 2.70 GHz

Detalhamento arquitetural:

O Intel Core i5-7200U trata-se de um processador dual-core com arquitetura Kaby Lake. Ele dispõe de dois núcleos de CPU com freqüência entre 2,5 e 3,1 GHz e integra HyperThreading para conseguir funcionar com até 4 threads ao mesmo tempo, em termos arquiteturais, a Intel usa basicamente a mesma microarquitetura em comparação com o Skylake (que são os processadores Intel Core de 6ª geração, que se utiliza, inclusive, do mesmo processo de fabricação 14 nm), de forma que o desempenho por MHz não difere. A Intel apenas retrabalhou a tecnologia Speed Shift para ajustes dinâmicos mais rápidos de voltagens e clocks, e o processo aprimorado de 14 nm permite frequências muito mais altas combinadas com melhor eficiência do que antes

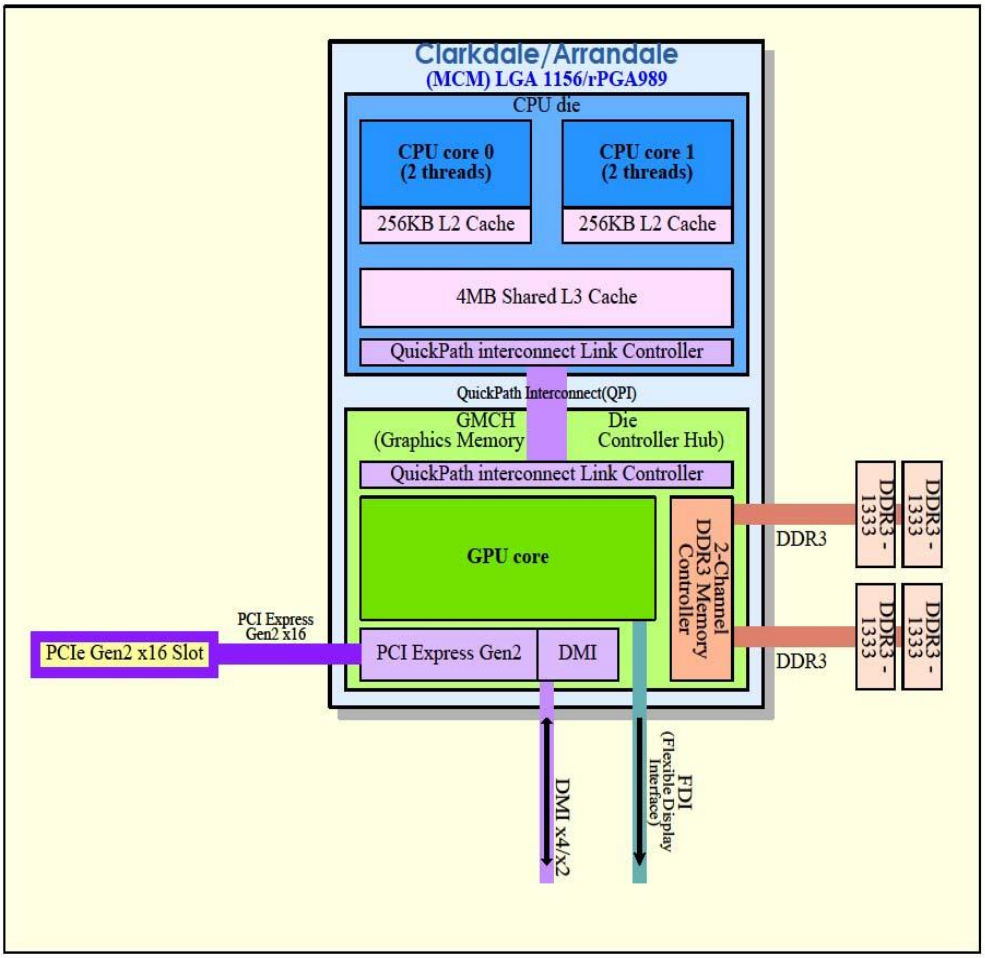
A seguir algumas representações gráficas de como é esse processador internamente:

Visão geral

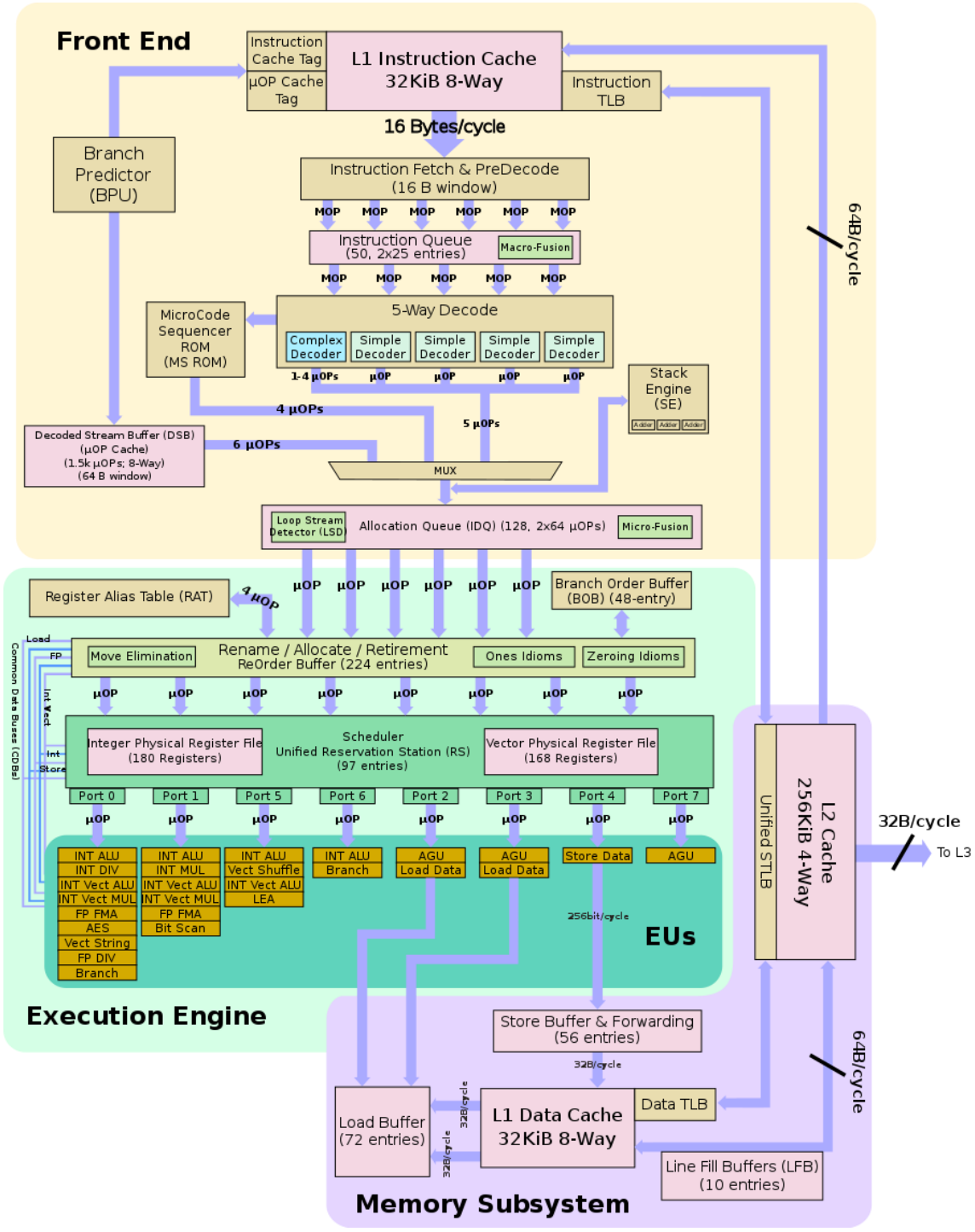


Visão mais detalhada

Clarkdale/Arrandale Architecture



Visão do core individualmente



Podemos ver pelas imagens que a idealização da construção arquitetural é para ele possui 2 cores (ou 2 processadores) e 4 threads, ou seja, 2 estados arquiteturais por processador, sendo esses 2 processadores físicos e 4 processadores lógicos auxiliados por memórias cache de menor tamanho se comparada a “L3” da imagem acima, que é uma memória de pouco espaço e de rápido acesso para auxiliar e volatilizar esses processadores. Em termos maiores ela é usada pela CPU com o objetivo de reduzir o tempo médio de acesso aos dados armazenados na memória, que é a outra parte (caixote) mostrada na imagem.

A parte voltada para a parte gráfica da memória também possui uma core, de maior tamanho, para administrar a parte gráfica, e sua construção é pensada para integrar essas 2 partes, sendo que a parte do processador utiliza mais do conceito de paralelismo pois faz-se necessário maior otimização dessa parte, mais necessariamente a CPU core 0 e a CPU core 1 da imagem acima, que por sua vez vão se utilizar de 2 threads paralelas (Por conta de que nesse processador se utiliza HyperThreading) para acelerar e otimizar ainda mais o tempo de processamento.

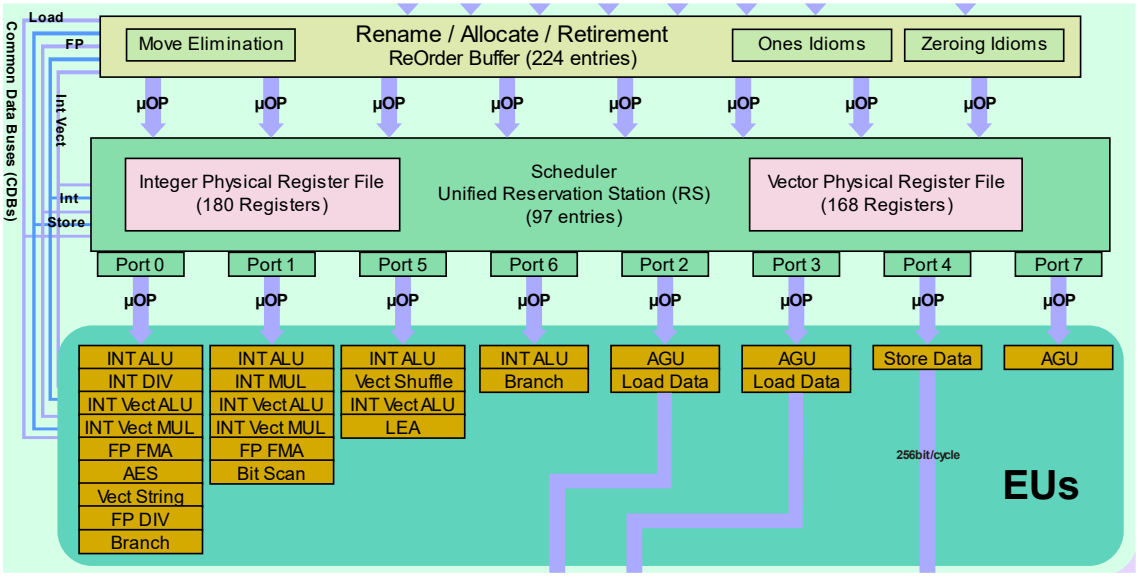
A Intel neste processador utiliza de uma tecnologia chamada de Enhanced Intel SpeedStep e Intel Speed para otimização e evitar desperdício de energia desnecessariamente. Quando o processador não está executando o código, ele fica ocioso. Um estado ocioso de baixa energia é definido por ACPI como um C-state. Essa tecnologia possibilita que todos os núcleos IA do processador ativo compartilham a mesma frequência e voltagem. Em um processador multicore, o estado P de frequência mais alta solicitado entre todos os IA núcleos ativos é selecionado.

Técnicas aplicadas:

No meu processador é utilizado Multithreading, mais especificamente HyperThreading, que é um caso de Multithreading proposto pela Intel.

Como vimos, temos 2 threads por processador, isso tem o objetivo da melhor utilização dos recursos, pois é compartilhado entre as threads unidades de processamento e de armazenamento, e o seu funcionamento e execução vai ser paralelo, permitindo ainda mais otimização do tempo do processado, versões maiores do processador podem executar dois threads em cada um de seus núcleos. Isso significa que cada thread obtém apenas metade dos recursos

Representação gráfica do pipeline superescalar do processador:



Por se tratar de uma arquitetura Kaby Lake, temos 14 estágios de pipeline, 16 se contarmos fetch/retire, e o tipo de pipeline é o **SuperEscalar**, e o pipeline dessa arquitetura é idêntico ao

do SkyLake, que por sua vez é resultado da evolução de outras arquiteturas, como o Nehalem pipeline e o Sandy Bridge pipeline (porém nota-se que que o pipeline do Sky e Kaby Lake é mais largo que os seus antecessores), ou seja, podemos ver aí uma evolução linear nos pipelines projetos pela Intel, que vão de acordo com a lógica de Tick Tock, que é um modelo adotado pela fabricante de chips Intel Corporation em 2007 para acompanhar todas as mudanças de microarquitetura com o processo de encolhimento da tecnologia anterior e o pipeline segue nessa ideia, ou seja, cada pipeline de uma arquitetura é estritamente baseado na arquitetura da geração anterior.

Ele é projetado para uma taxa de transferência de quatro instruções por ciclo de clock, todas as partes do pipeline são compartilhadas entre dois threads nesses modelos de CPU que podem ser executados dois threads em cada núcleo. Cada thread obtém metade do rendimento total quando dois threads estão rodando no mesmo núcleo

Visão geral do funcionamento do pipeline do kaby lake:

Esse pipeline pode ser dividido em três áreas: front-end, back-end ou mecanismo de execução e o subsistema de memória.

O objetivo do front-end é alimentar o back-end com um fluxo suficiente de operações que ele obtém decodificando instruções vindas da memória.

O front-end tem dois caminhos principais: o caminho do cache µOPs e o caminho legado. O caminho legado é o caminho tradicional pelo qual as instruções x86 de comprimento variável são buscadas no cache de instrução de nível 1, enfileiradas e, consequentemente, são decodificadas em µOPs mais simples e de comprimento fixo . O caminho alternativo e muito mais desejado é o caminho do cache µOPs, pelo qual um cache contendo µOPs já decodificados recebe um hit que permite que os µOPs sejam enviados diretamente para a fila de decodificação. Independentemente do caminho que uma instrução tome, ela acabará chegando à fila de decodificação. O IDQ representa o fim do front-end e a parte em ordem da máquina e o início do mecanismo de execução que opera fora de ordem.

No back-end, as microoperações visitam o buffer de reordenamento . É lá onde ocorre a alocação, renomeação e retirada de registro . Nesta fase, uma série de outras otimizações também são feitas. Do buffer de reordenamento, µOPs são enviados para o agendador unificado. O planejador tem várias portas de saída, cada uma conectada a um conjunto de unidades de execução diferentes. Algumas unidades podem realizar operações básicas de ALU, outras podem fazer multiplicação e divisão, com algumas unidades capazes de operações mais complexas, como várias operações vetoriais.

O escalonador é efetivamente responsável por enfileirar os µOPs na porta apropriada para que possam ser executados pela unidade apropriada. Alguns µOPs lidam com acesso à memória (por exemplo, carregar e armazenar). Eles serão enviados em portas de agendador dedicadas que podem executar essas operações de memória. As operações de armazenamento vão para o buffer de armazenamento, que também é capaz de realizar o encaminhamento quando necessário. Da mesma forma, as operações de carregamento vêm do buffer de carregamento. Kaby lake apresenta um cache de dados de nível 1 dedicado de 32 KiB e um cache de instrução de nível 1 de 32 KiB dedicado. Ele também possui um cache L2 de 256 KiB privado de núcleo que é compartilhado por ambos os caches L1.

Cada núcleo desfruta de uma fatia de um terceiro nível de cache que é compartilhada por todo o núcleo. Para o Kaby Lake, existem dois núcleos ou quatro núcleos conectados em um único chip.

Otimizações e achatamentos no Kaby Lake:

Assim como as outras questões, as otimizações do Kaby Lake são extremamente similares a do SkyLake, que por sua vez tem uma série de otimizações que executa antes de inserir a parte fora de ordem e renomear. Três dessas otimizações incluem Eliminação de movimento e Zerar idiomas, e outros idiomas. Uma Eliminação de movimento é capaz de eliminar movimentos de registro para registro (incluindo movimentos encadeados) antes da contabilidade no ROB, permitindo que esses µOPs economizem recursos e os elimine totalmente. Movimentos eliminados têm latência zero e são totalmente removidos do pipeline. Essa otimização nem sempre é bem-sucedida; quando falha, os operandos simplesmente não estavam prontos. Em média, essa otimização quase sempre é bem-sucedida (mais de 85% na maioria dos casos). A eliminação de movimento funciona em todos os registros inteiros GP de 32 e 64 bits, bem como em todos os registros vetoriais de 128 e 256 bits.

Uma segunda otimização comum realizada no Kaby Lake na mesma época é a eliminação de Zeroing Idiomas. Vários idiomas de zeragem comuns são reconhecidos e, consequentemente, eliminados da mesma maneira que as eliminações de movimento são realizadas. O Kaby Lake reconhece instruções como XOR, PXOR e XORPS como expressões idiomáticas de zeragem quando os operandos de origem e destino são os mesmos. Essas otimizações são feitas na mesma taxa da renomeação durante a renomeação (a 4 µOPs por ciclo) e o registro é simplesmente definido como zero.

Exemplo:

Exemplo de zeragem de idioma :

```
xor eax, eax
```

Essa instrução não apenas é eliminada no ROB, mas na verdade é codificada como apenas 2 bytes `31 C0` contra os 5 bytes para os quais é codificado como . `mov eax, 0x0` `b8 00 00 00 00`

Os idiomas uns é outro idioma de quebra de dependência que pode ser otimizado. Em todas as várias instruções PCMPEQx que executam a comparação compactada, o mesmo registro consigo mesmo sempre define todos os bits como um. Nesses casos, enquanto o µOP ainda precisa ser executado, as instruções podem ser agendadas o mais rápido possível porque o estado atual do registro não precisa ser conhecido.

Referências e sites de onde eu tirei essas informações e imagens:

(sites de especializados)

https://www.notebookcheck.net/Intel-Core-i5-7200U-Notebook-Processor.172250.0.html

https://www.legitreviews.com/intel-7th-gen-kaby-lake-processor-architecture-details-released_185842

https://www.cin.ufpe.br/~if674cc/aulas/AulaInfraHW-Superscalar.pdf

https://edisciplinas.usp.br/pluginfile.php/3418115/mod_resource/content/16/15aula%20-%20Evolucao%20Arq%20Intel%20-%20parte%202.pdf

https://www.agner.org/optimize/microarchitecture.pdf

(Site que contém todos os Datasheets que eu peguei da Intel)

https://www.intel.com/content/www/us/en/design/products-and-solutions/processors-and-chipsets/kaby-lake-s/technical-library.html?grouping=EMT_Content%20Type&sort=title:asc