4.2 Регистри – Собствена памет на процесора – с най-малък капацитет и най-високо бързодействие. Временно съхраняват операндите, резултатите, текущо изпълняваните инструкции и др.

2. Микроархитектура концепции, реализирани в съвременните процесори

2.1 Развитие на системата от инструкции (Instruction set)

A) Основни класове инструкции

* Трансфер на данни – регистри, памет
* За аритметични операции
* За логически операции
* За управление на изпълнението – преход, цикъл, прекъсване
* За обработка на символни данни
* За B/И

Б) Мултимедийни разширения на системата от инструкции

* Необходимост – ускорени изчисления, свързани с обработката на видео и аудиоизображения, 3D графика, компресия/декомпресия – компютърни игри, комуникация, Internet, мултимедия, видеоконференции и др.
* **Идея –** едновременно (групова) обработка на няколко елемента от данните с една инструкция (технология SIMD – Single Instruction Multiple Data):
  + **Изображение – група пиксели**
  + **Звук – група стойности на амплитудата**
* **Особености:**
  + **Нови типове (пакетирани) данни и нови регистри – пакетират все по-голям брой елементи от данни, т.е. имат по-голяма разрядност**
  + **Пресмятане с насищане: резултат от арит. Операция – към максимална/минимална стойност**
* **Разширения:**

**Intel**

* **MMX (Multimedia eXtensions) – Pentium MMX (1997 г.)**
* **SSE (Streaming SIMD eXtensions) – Pentium III (1999 г.)**
  + **SSE2 – Pentium 4 (2000 г.)**
  + **SSE3 – Pentium 4 (2004 г.)**
  + **SSE4 – 4.1 – Pentium 4 (2006 г.) и 4.2 – Core i (2008 г.)**
* **AVX (Advanced Vector Extensions), 2008 г.**
  + **Intel – Sandy Bridge и Ivy Bridge**
  + **AMD – Bulldozer и Piledriver**
* **AVX2 – Intel – Haswell (2013 г.)**

**AMD**

* **3DNow!, Enhanced 3DNow! (3DNow+) и SSE 4.a**
* **От 2010 г. – отказ от поддържане на своите разширения, с. Изкл. На SSE 4.a**

**Други разширения**

**2.2 Конвейерност (Pipelining)**

**A) Цикъл на инструкцията**

* **Адресиране и четене на инструкция (Instruction Fetch – IF)**
* **Декодиране на инструкция (Instruction Decoding – ID)**
* **Изпълнение на инструкция (Instruction Execution – EX)**
* **Съхранение на резултата (Write Back – WB) – регистър/памет.**

**Б) Идея на конвейертизацията - едновременно изпълнение на различни фази от съседни инструкции**

**Пример за конвейер**

**Идеалния конвейер**

1. **Фазите на продължителност в рамките на един такт**
2. **Едновременно изпълняваните инструкции са независими**
3. **Инструкции са линейни**

**Необходими условия**

**Първи конвейер – Intel 80486**

**Основна характеристика на конвейерите – дълбочина (брой фази)**

* **Intel 80486 – 5 фази**
* **Pentium 4 – 20 фазен (0.18**

**Конфликт в реалните конвейери**

**Конфликт – ситуация в конвейра, която пречи изпълнението на поредната инструкция да стане в предназнаение**

***Времедиаграма на примерен конфликт***

**Извличането на ,,виолетова” инструкция заема 2 такта**

**B1.Ресурсни конфликти**

**Причина – конкуренцията на две или повече инструкции за едновременно заемане на един и същ ресурс, напр. Определен регистър.**

**B2. Конфликти по управление**

**Причина – нарушаване естествения ред на изпълнение на инструкциите в програмата – прехода, цикли, прекъсване и др.**

**Типичен пример – инструкции за условен преход:**

* **Изпълнението на прехода – на фаза “изпълнение”**
* **Необходимост от изпразване на конвейира при преход.**

**B3. Конфликти по данни**

**Причина – изпълнение на една инструкция зависи от резултата от изпълнението на предходна инструкция.**

**Типичен пример – инструкция i+1 се опитва да прочете операнд, който се формира след изпълнението на предходната инструкция i, а той все още не е готов (инструкция i+1 ще получи старата стойност на операнда).**

**2.3 Суперскаларност**

**A) Същност**

* **Паралелно изпълнение на няколко инструкции чрез няколко конвейира, най-често реализации фази EX (изпълнение) – наличност за няколко ALU или FPU.**
* **Реализация – чрез добавяне на ресурси и/или използване на назаети в момента ресурси с друго предназначение, напр. Използване на FPU за изпълнение на някои операции с фиксирана запетая, изпълнявани стандартно от ALU.**

**Б) Основни параметри за характеризиране на скаларността:**

* **Брой конвейри за целочислена (ALU) и плаваща (FPU) аритметика**
* **Равнопоставеност на еднотипните конвейри:**
  + **Равнопоставени – всеки конвейр изпълнява както прости инструкции (изпълняват се хардуерно), така и сложни (свеждат се до поредица о микроинструкции – ROM)**
  + **Неравнопоставеност**

**Дълбочина на конвейерите**

**Съвместивост (едновременност) в работата на конвейерите за целочислена и плаваща аритметика**

**Напр. Pentium 4**

* **4 конвейра за аритметика с фиксирана запетая (ALU) и 2 конвейера**

**Примерна архитектира – AMD Bulldozer**

**2.4 Техника за повишаване производителността на конвейерите**

**A) Прогнозиране на условните преходи (Branch Prediction)**

**Специален модул (блок за прогнозиране) определя най-вероятното поведение на програмата – да се изпълни преходът или да продължи линейното изпълнение на инструкциите. В съответствие с прогнозата, извличането на инструкциите продължава, без да се изчаква проверката на условието за преход (на фаза Изпълнение).**

* **При потвърждаване на прогнозата, работата на конвейера продължава.**
* **При грешна прогноза целият конвейер се изпразва, междинните резултати от всяка фаза се изчисляват, след което конвейерът се зарежда с инструкциите от другия клон на програмата.**

**Според Intel е възможно да се предскажат с вероятност 90-95 % над 1000 прехода.**

**Б) Спекулативно изпълнение, или изпълнение по предложение (Speculative Execution)**

**Техника е известна още под името “изпълнение в аванс”:**

**Представлява развитие на идеята за прогнозиране на преходите –**

**Предсказваните инструкции не само се извличат, но и се изпълняват. Резултатите от обработката временно се съхраняват в междинен буфер – до потвърждаване на прогнозата.**

* **При потвърждаване на прогнозата (фаза Изпълнение) резултатите се записват (Write Back) в регистрите с общо назначение или в паметта; изпълнението на извлечените инструкции продължава.**
* **При грешна прогноза резултатите се изтриват от буфера, целият конвейер се изпразва, след което започва да се зарежда с инструкциите от другия клон на програмата.**

**В) Неподредено изпълнение на инструкциите (Out-of-Order Execution)**

* **Същност**

**Инструкцията се изпълняват не в реда, в който следват в програмата (In-Order execution), а в зависимост от готовността си за изпълнение, т.е. необходимите ресурси или в кеша на процесора. След неподреденото изпълнение на инструкциите, процесорът пренарежда и записва резултатите в съответствие с реалната последователност на инструкциите.**

* **Резултат**

**Избягват се престоите на процесора в тези случаи, в които данните, необходими за изпълнение на поредната инструкция не са готови.**

* **Нови блокове – блок за анализ и изтегляне на готовите за изпълнение инструкции (Retirement Unit) и буфер за пренареждане на резултатите (Reorder buffer).**