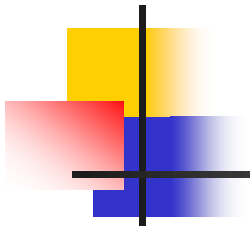
GRAPHICS & MEDIA LAB
VIDEO GROUP

Обзор некоторых современных SoC

Юрий Бердников

Video Group
CS MSU Graphics & Media Lab



Содержание

- Оптимизации PNX
- Texas Instruments DaVinci
- Stretch
- NVIDIA Tegra
- ARM11
- Сравнение

Содержание

- Оптимизации PNX
  - Фиксированная точка
  - Restricted pointers
  - Оптимизация расположения данных
  - Личный опыт
- Texas Instruments DaVinci
- Stretch
- NVIDIA Tegra
- ARM11
- Сравнение

Оптимизации PNX

Фиксированная точка

- Конвертация из float

  A = (int) (a * (1<<N));

  Latency = 6

- Конвертация в float

  a = (float) A*(1.0/(1<<N));

  Latency = 6

- Сложение и вычитание

  A=B+C; D=E-F;

  fixedLatency = 1

  floatLatency = 3

Оптимизации PNX

Фиксированная точка

- Умножение

  Степень 2: A=B<<C

  fixedLatency=1

  Целое число: A=B*C

  fixedLatency=3

  Общий случай:

  A=IMULM(B, C)<<NORM

  fixedLatency=4

  floatLatency=3

- Деление

  Степень 2: A=B>>C

  fixedLatency=1

  Целое число: A=B/C

  fixedLatency=23, fixedRecovery=17

  Общий случай:

  A=(int)((corr*B/C)

  fixedLatency=26, fixedRecovery=17

  floatLatency=17, floatRecovery=17

Оптимизации PNX

Restricted pointers

compiler must assume pointers to a, b, c could overlap

```
void unrolled_convolution(char *a, char *b, int *c)
                                        ...
        c[0] = b[0]*a[0]+b[1]*a[-1]+ b[2]*a[-2]+b[3]*a[-3]
               +b[4]*a[-4]+b[5]*a[-5]+b[6]*a[-6]+b[7]*a[-7];
        c[1] = b[0]*a[1]+b[1]*a[0]+b[2]*a[-1] + b[3]*a[-2]
               +b[4]*a[-3]+b[5]*a[-4]+b[6]*a[-5]+b[7]*a[-6];
                                        ...
```

so this second assignment could be affected by...

...this first one

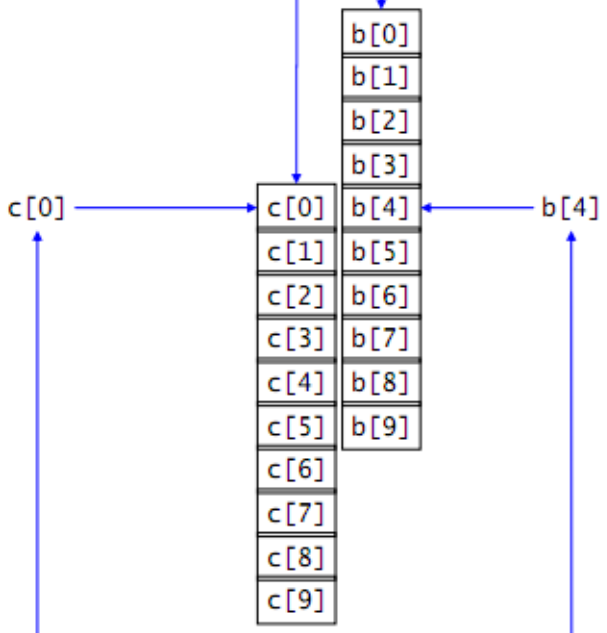
```
restrict_direct_convolution(
   char * restrict a, char * restrict b, char * restrict c
) {
```

restrict keyword tells compiler the pointers don't overlap

in C, arrays are accessed by pointers

```
c[k] = *(&c[0] + sizeof(c)*k)
```

so arrays may overlap



c[0] may point to the same object as b[4]

Оптимизации PNX

Оптимизация расположения данных

- Характеристики кэша PNX1500
  - Объём – 16 КБ
  - Блок – 64 байта
  - Регистр<->кэш – 3 цикла
  - Кэш<->память – 40 циклов
- MPEG-2 декодер
  - Задача компенсации движения
  - Полупиксельная точность
  - Блок 17x17 для Y, 8x8 для U и V
  - Оценка эффективности – количество обращений кэша к памяти на блок, худший случай

Оптимизации PNX

Оптимизация расположения данных

- Обращения к памяти
  - Y – 34 обращения
  - U – 16 обращений
  - V – 16 обращений
  - Итого – 66 обращений
- Использование канала
  - Y: 13.1%
  - UV: 7.9%

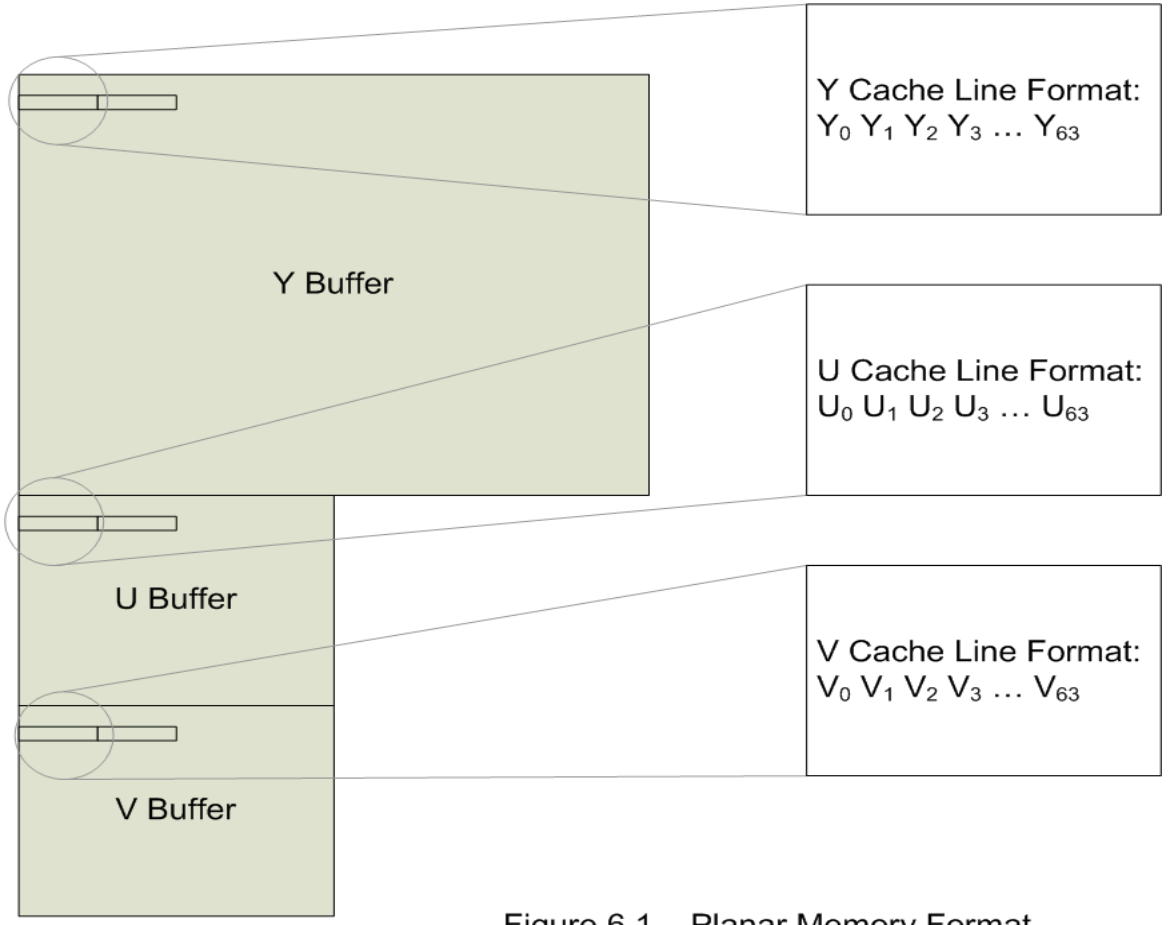


Figure 6.1 – Planar Memory Format

Оптимизации PNX

Оптимизация расположения данных

- Обращения к памяти
  - Y – 34 обращения
  - UV – 16 обращений
  - Итого – 50 обращений
- Использование канала
  - Y: 13.1%
  - UV: 15.8%

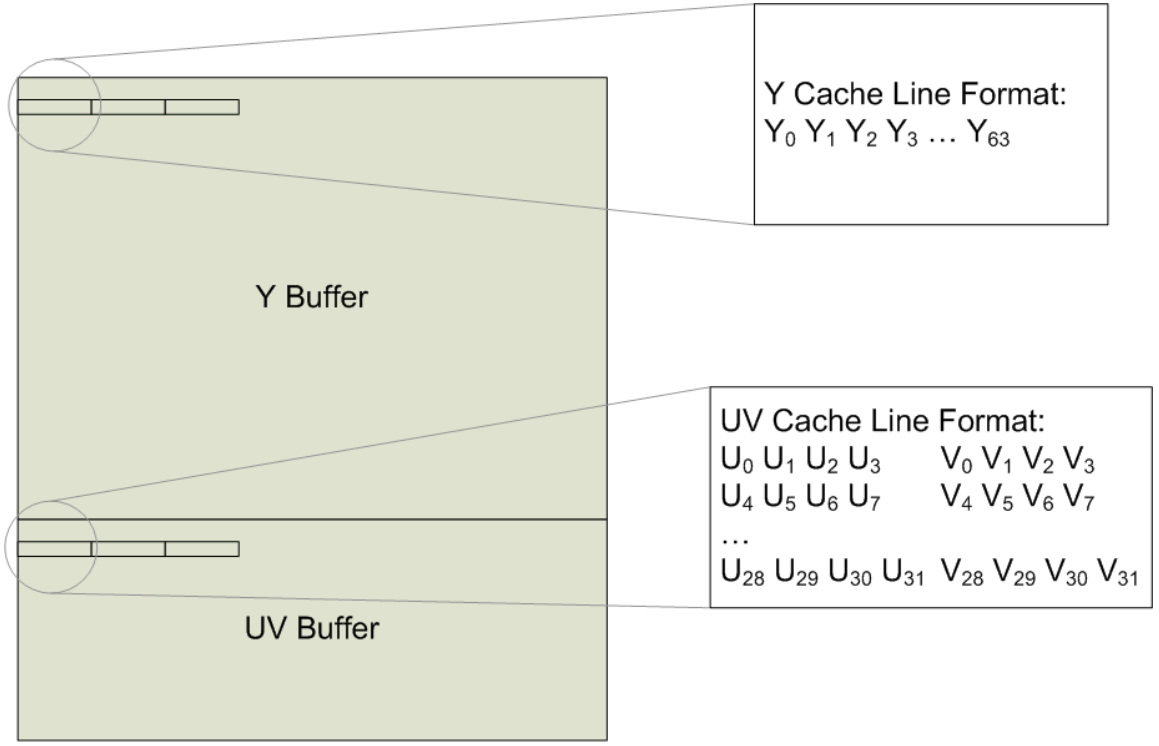


Figure 6.2 – Quad Semi-Planar Memory Format

Оптимизации PNX

Оптимизация расположения данных

- Обращения к памяти
  - Y – 9 обращений
  - UV – 8 обращений
  - Итого – 17 обращений
- Использование канала
  - Y: 50.2%
  - UV: 31.6%

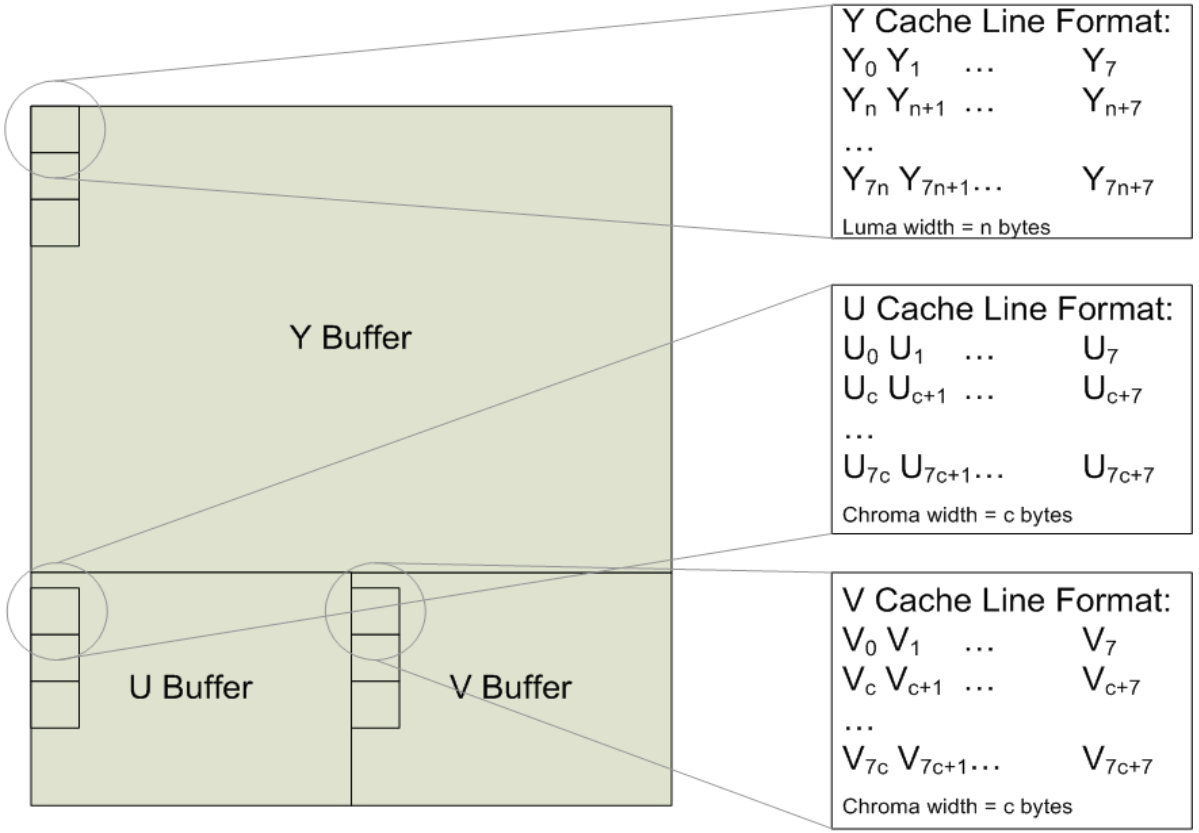


Figure 6.3 – 8x8 Tiled Memory Format

Оптимизации PNX

Из личного опыта

- Профилировщик влияет на оптимальность операций!

```
//степень развёртывания - 2
for (x=0; x<ext_width; x+=2){
        Y[x  ]=SplA[x]*(Y[x]-128)+128+SplB[x];
        Y[x+1]=SplA[x+1]*(Y[x+1]-128)+128+SplB[x+1];
}
```

Зависимость FPS от профилировщика и степени развёртывания циклов

Профилировщик	Степень развёртывания цикла			
	3	4	5	6
Выкл.	4.52 FPS	4.65 FPS	**4.71 FPS**	4.43 FPS
Выкл.	8.01 FPS	**8.24 FPS**	8.06 FPS	7.54 FPS

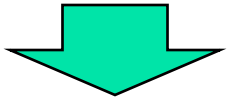
Работа всего фильтра, приведённый фрагмент занимает ~10% циклов

Оптимизации PNX

Из личного опыта

- Однообразные инструкции следует группировать

```
for (x=0; x<ext_width; x+=2){
        Y[x]=SplA[x]*(Y[x]-128)+128+SplB[x];
        Y[x+1]=SplA[x+1]*(Y[x+1]-128)+128+SplB[x+1];
}
```



```
for (x=0; x<ext_width; x+=2){
        tmp[0]=Y[x]-128;
        tmp[1]=Y[x+1]-128;
        tmp[0]*=SplA[x];
        tmp[1]*=SplA[x+1];
        tmp[0]+=128;
        tmp[1]+=128;
        Y[x]=tmp[0]+SplB[x];
        Y[x+1]=tmp[1]+SplB[x+1;
}
```

Прирост скорости – 1.5 раза!

Оптимизации PNX

Из личного опыта

- Компилятор циклы разворачивает не всегда
- Конструкции вида

```
for (x=0; x<ext_width; x+=4){
        Y[x] = <some expression>
        Y[x+1] = <some expression>
        Y[x+2] = <some expression>
        Y[x+3] = <some expression>
}
```

Медленнее, чем

```
for (x=0; x<ext_width; x+=4){
        Y[0]= <some expression>
        Y[1]= <some expression>
        Y[2]= <some expression>
        Y[3]= <some expression>
        Y+=4;
}
```

Оптимизации PNX

Из личного опыта

- Inline для больших функций может замедлить работу
- Следует избегать обращения по вычисляемым адресам

```
for (x=0; x<ext_width; x+=4){
        Histogram[Y[x]]++;
        Histogram[Y[x+1]]++;
        Histogram[Y[x+2]]++;
        Histogram[Y[x+3]]++;
}
```

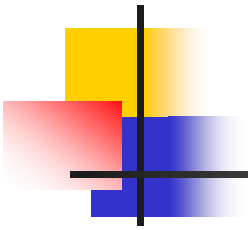
~7.8 циклов/пиксель!

- Конструкции вида

```
[vars for alg1]
[vars for alg2]
[alg1]
[alg2]
```

Медленнее, чем

```
{
  [vars for alg1]
  [alg1]
}
{
  [vars for alg2]
  [alg2]
}
```



Содержание

- Оптимизации PNX
- Texas Instruments DaVinci
  - Введение
  - Устройство
  - ARM256
  - C64x+
  - Программирование
- Stretch
- NVIDIA Tegra
- ARM11
- Сравнение

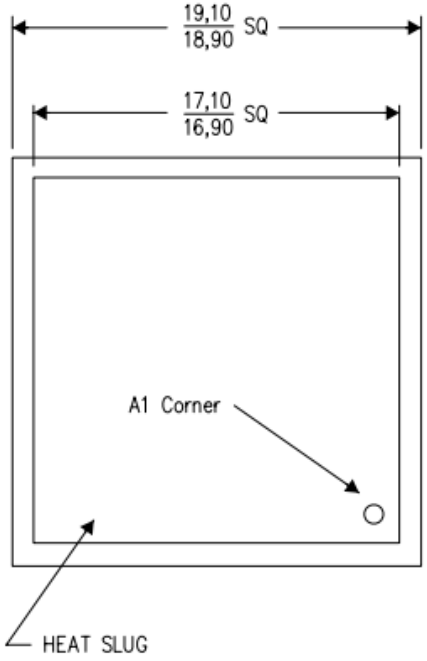
Texas Instruments DaVinci

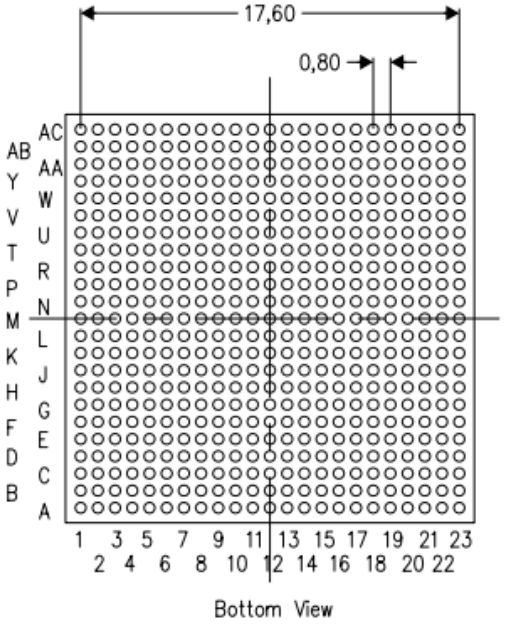
Введение

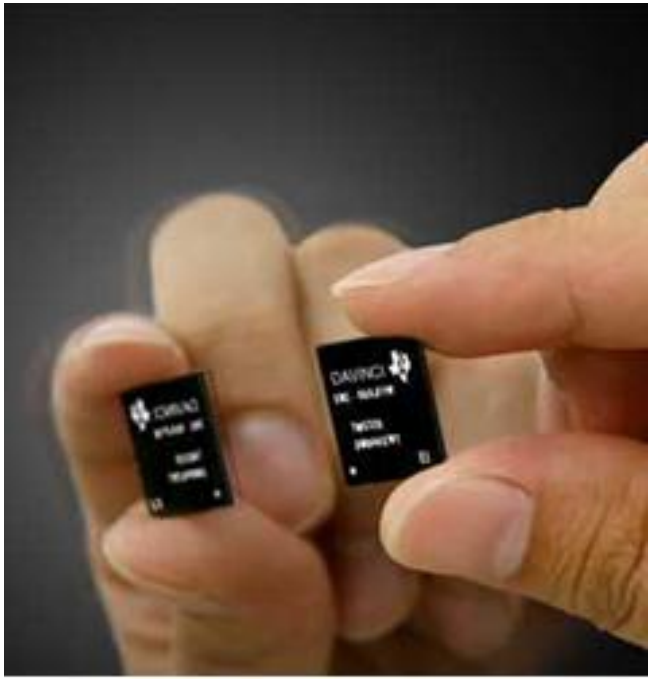
- Широкая линейка SoC
- Ядро – ARM926 и/или C64x+
- 135 MHz – 900 MHz
- Различный набор сопроцессоров
- Область применения - обработка видео
- 10$ - 90$

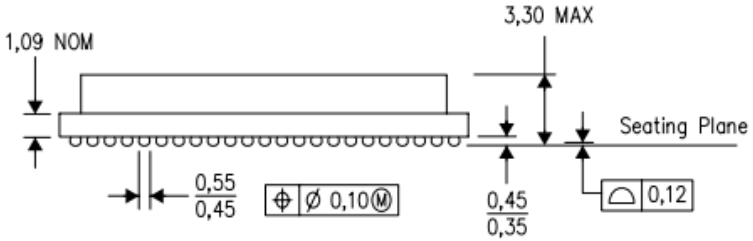
Texas Instruments DaVinci

Физические параметры









Texas Instruments DaVinci

Сравнительная таблица моделей

DaVinci™ Processors: Tuned for Digital Video End Equipments

DaVinci Processor	CPU	MHz	Capture/ Display
DM335*	ARM926	135, 216, 270	Capture/Display
DM355*	ARM926**	135, 216, 270	Capture/Display
DM6467	C64x+™/ARM926†	594/297	Capture/Display
DM648	C64x+	720, 900	Capture/Display
DM647	C64x+	720, 900	Capture/Display
DM6446*	C64x+/ARM926	600/300	Capture/Display
DM6443	C64x+/ARM926	600/300	Display
DM6441*	C64x+/ARM926	512/256	Capture/Display
DM6437	C64x+	400, 500, 600	Capture/Display
DM6435	C64x+	400, 500, 600	Capture
DM6433	C64x+	400, 500, 600	Display
DM6431	C64x+	300	Capture

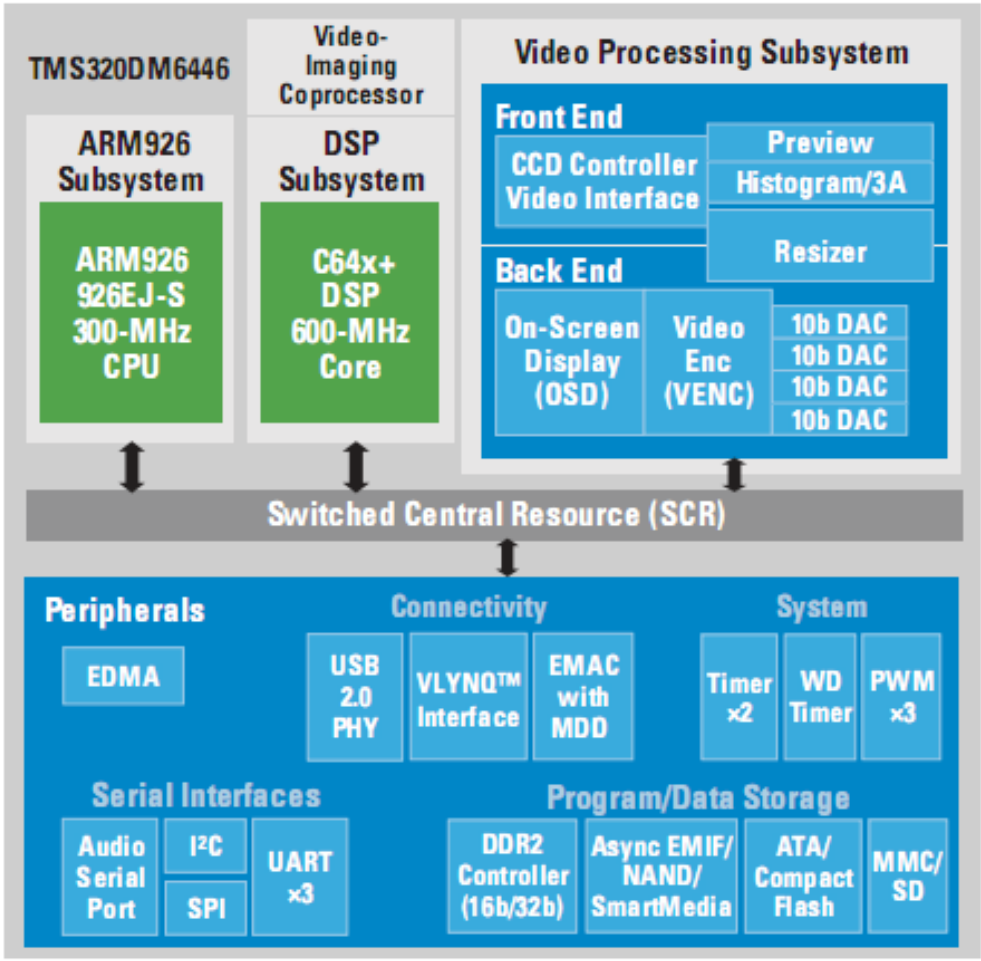
*Includes video imaging co-processor **Includes MPEG-4/JPEG co-processor
†Includes DaVinci High-Definition video/imaging co-processors

Texas Instruments DaVinci

Пример процессора

- TMS320DM6446-513
- ARM Frequency - 256.5 MHz
- DSP Frequency - 513 MHz
- VICP - 256.5 MHz
- Peak MMACS - 4104
- L1 - 112 KB (DSP),40 KB (ARM)
- L2 - 64 KB (DSP)

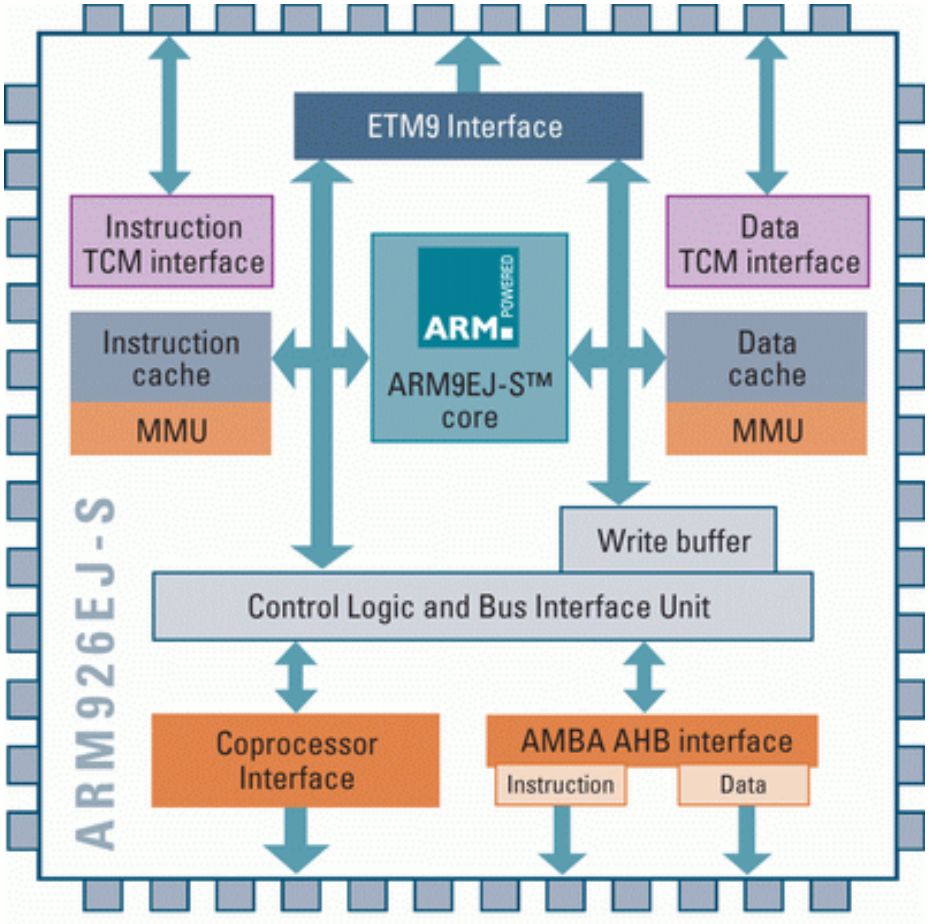
- Цена - 42$



Texas Instruments DaVinci

ARM926EJ-S

- RISC-архитектура
- Технология – 110 nm
- Низкое потребление
- Управляемый кэш
- Расширяемость за счёт сопроцессоров
- Только целочисленная арифметика
- Полный набор команд для нормальной работы ОС

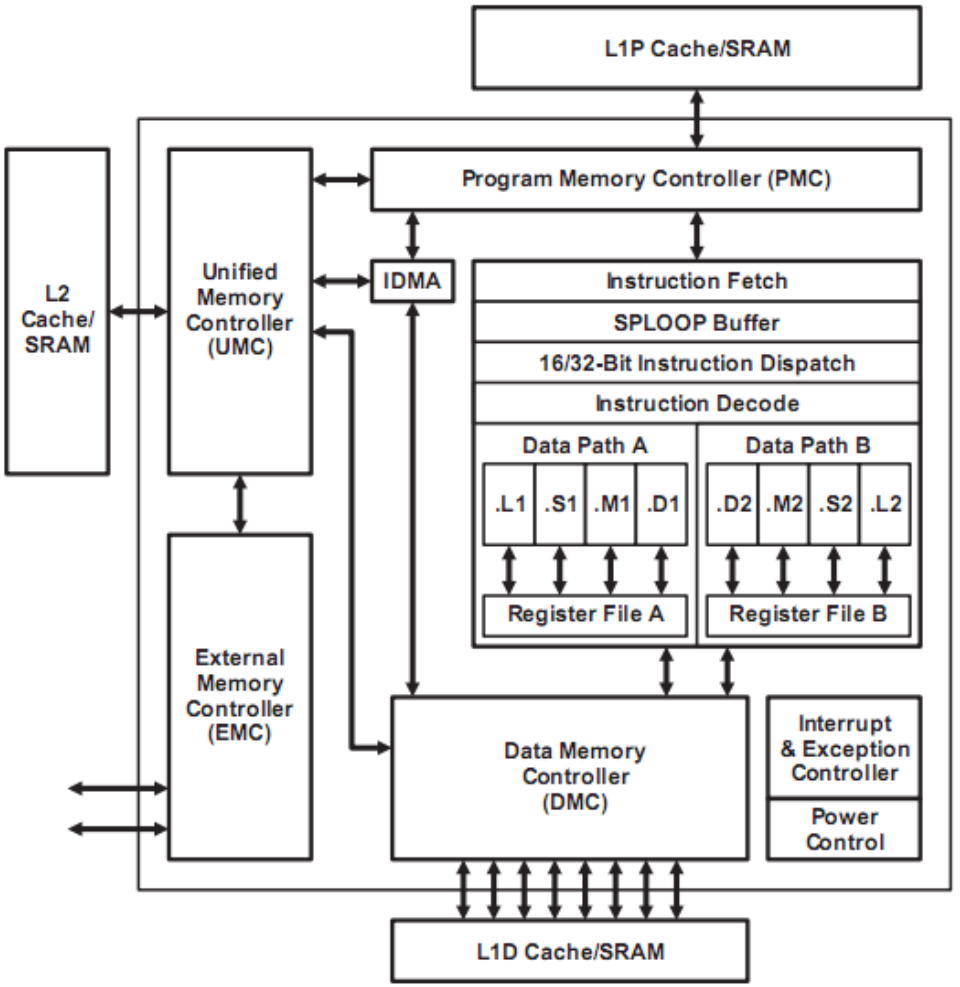


Texas Instruments DaVinci

C64x+

- 500-700 MHz
- 64 32-битных регистра
- 8 функциональных блоков
- 48 ALU
- 16 устройств умножения
- 2 потока данных (Data Path)
- Internal DMA
- 32 KB L1P cache
- 16 KB L1D cache
- 64 KB L2 cache

Figure 1-2. TMS320C64x+ DSP Block Diagram

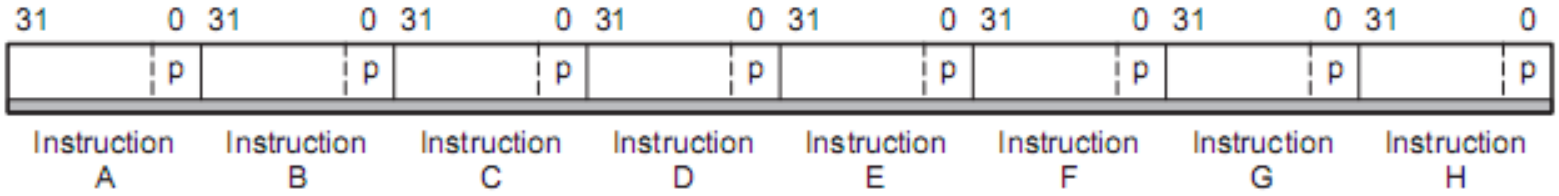


Texas Instruments DaVinci

C64x+

- Загрузка инструкций блоками по 256 байт
- От 8 до 14 инструкций в блоке
- Параллельное выполнение до 8 операций

Figure 3-1. Basic Format of a Fetch Packet

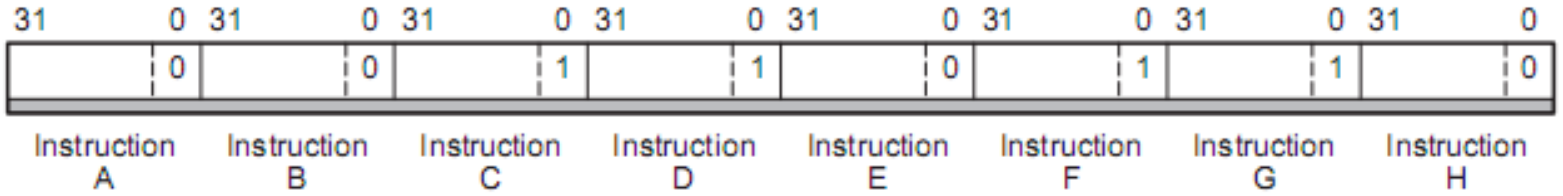


Texas Instruments DaVinci

C64x+

GRAPHICS & MEDIA LAB
VIDEO GROUP

Пример параллельного выполнения операций




```
;Так это выглядит на ассемблере

                Instruction A
                Instruction B
                Instruction C
        ||      Instruction D
        ||      Instruction E
                Instruction F
        ||      Instruction G
        ||      Instruction H
```

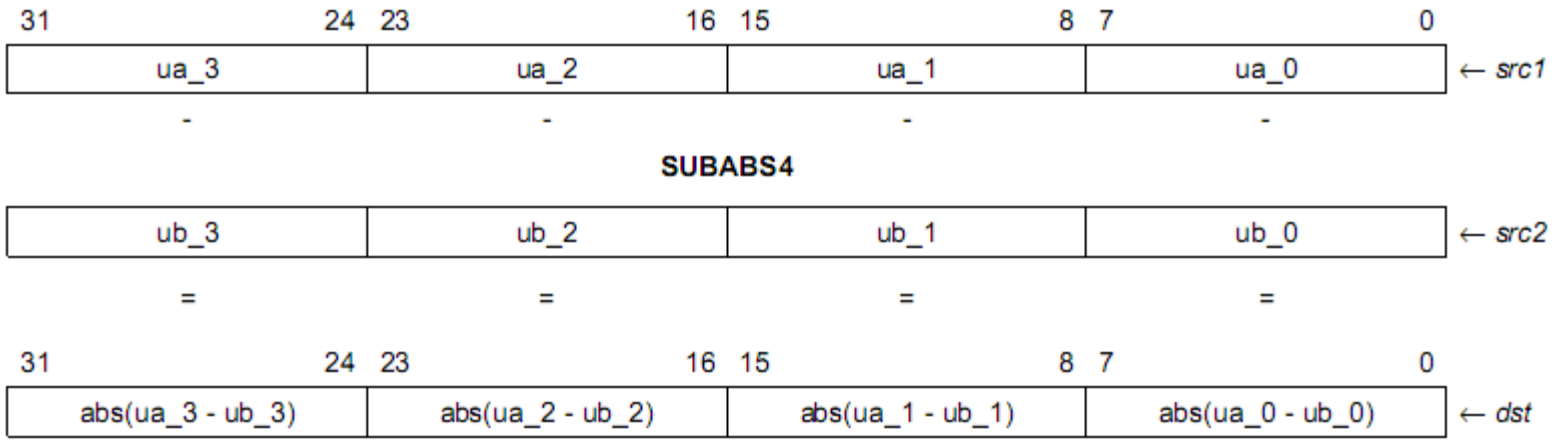
Шаг	Выполняемые инструкции
1	A
2	B
3	C D E
4	F G H

Texas Instruments DaVinci
C64x+

- «Отложенные» операции

```
MVD       .M2x    A0,    B0        ;
NOP                                 ;
NOP                                 ;
NOP                                 ; B0 = A0
```

- Операции, ориентированные на DSP



Texas Instruments DaVinci
C64x+

- «Условное» выполнение большинства операций

```
        [B0]    ADD     .L1     A1,A2,A3
||      [!B0]   ADD     .L2     B1,B2,B3
```

Ускорение до 10 раз!

- 40- и 64-битная арифметика
- Специальные команды для организации циклов
- Аппарат прерываний
  - Таймеры
  - Прерывания от АЦП/ЦАП
  - Прерывания от сопроцессоров

Texas Instruments DaVinci

C64x+

- Ориентация на работу по 4 байта
- Отсутствие операций деления
- Аппаратные функции для работы с алгоритмами коррекции ошибок
  - Поле Галуа Gf(8), полином $P(x)=x^3+x+1$
  - Коды Рида-Соломона

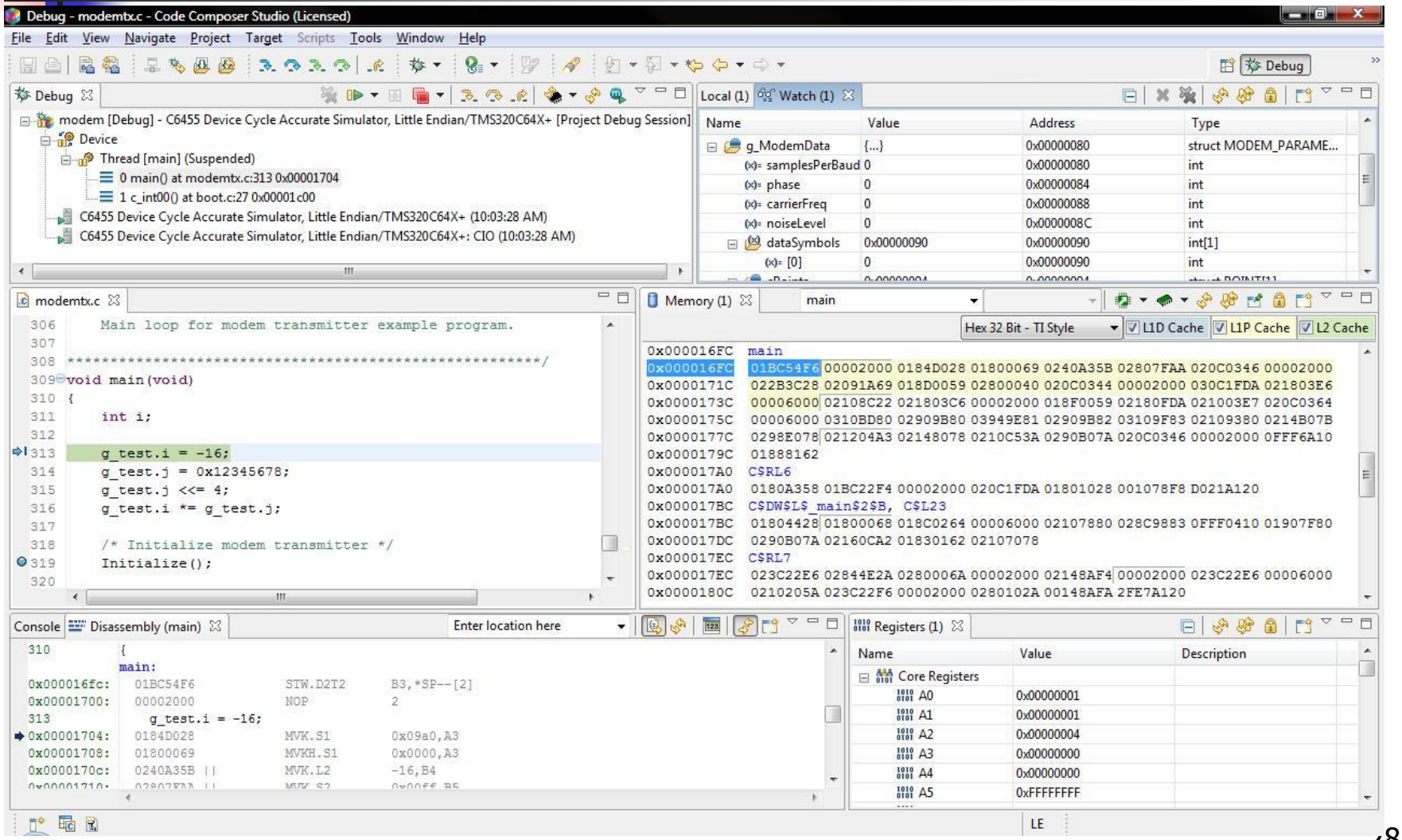
Texas Instruments DaVinci

Программирование

- Язык – C, C++ с дополнительными командами
- ОС - Windows, Linux, MacOS
- Code Composer Studio
  - IDE (На основе Eclipse)
  - Заголовочные и библиотечные файлы
  - Компилятор
  - Средства отладки
  - Программы эмуляции
  - Профилировщики

Texas Instruments DaVinci

Программирование. Code Composer Studio



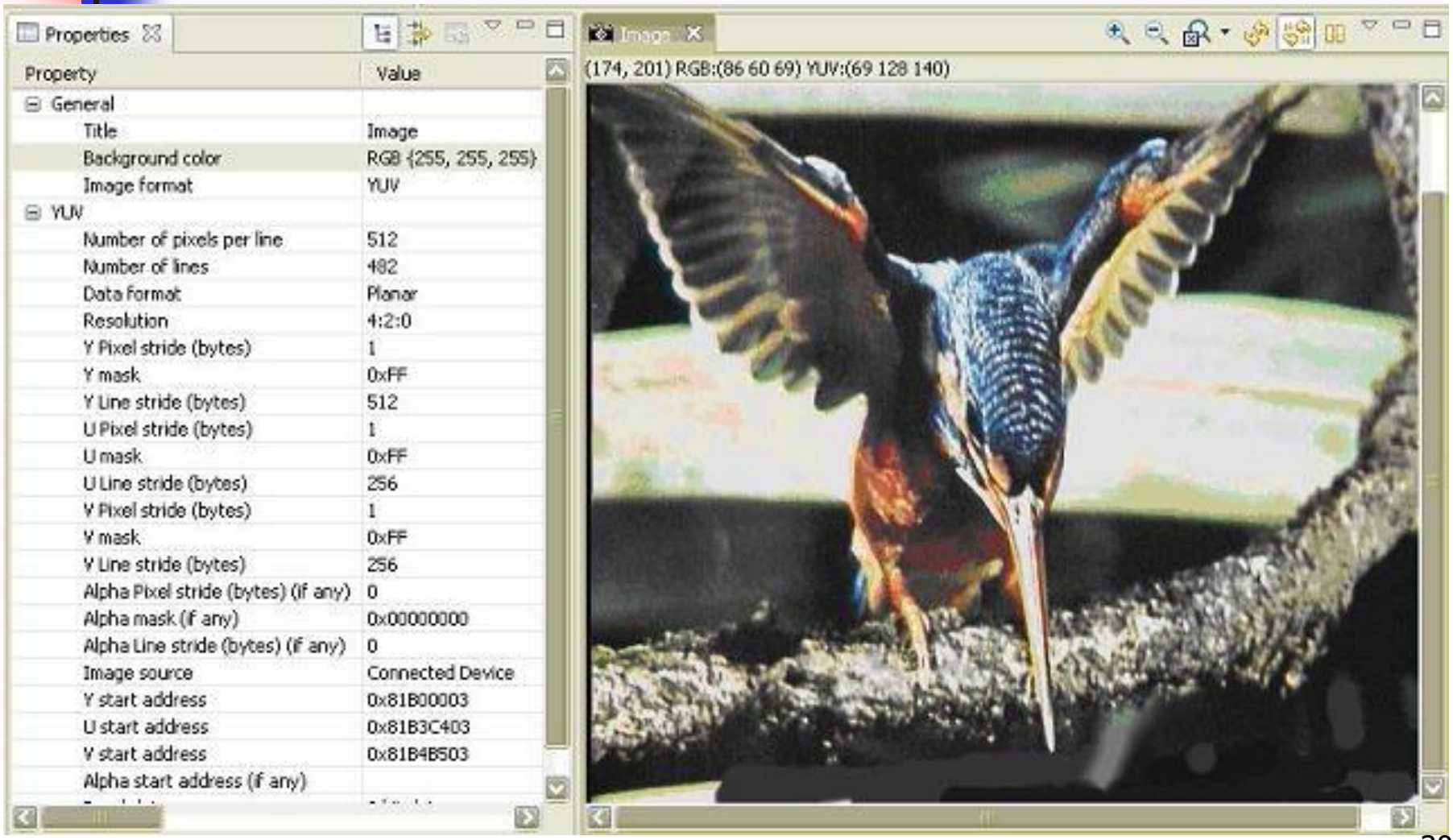
Texas Instruments DaVinci

Программирование.Code Composer Studio

- Особенности CCStudio
  - Независимость от платформы
  - Независимость от языка программирования
  - Поддержка скриптов
  - Оптимизирующий компилятор
  - Взаимодействие с DSP/BIOS
  - Средства для визуального контроля

Texas Instruments DaVinci

Программирование.Code Composer Studio



Texas Instruments DaVinci

Программирование. Hello, world!

```
//Simple HelloWorld program – only ARM
void main(int argc, char *argv[])
{
   printf("DaVinci Rulezzz!!!\n");
   fflush(stdout);
   return;
}
```

- ARM и DSP – разные вещи!
  - ARM – основной процессор, на нём работает ОС
  - DSP – сопроцессор
- кэш у каждого свой
- Низкая скорость при выполнении последовательных команд разными процессорами

Texas Instruments DaVinci

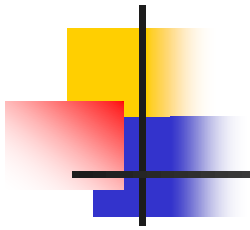
Программирование. Hello, world!

GRAPHICS & MEDIA LAB
VIDEO GROUP

Раздельные программы для ARM и DSP

```
//Simple HelloWorld program – DSP part
void main(int argc, char *argv[])
{
   printf("DaVinci Rulezzz!!!\n");
   fflush(stdout);
   return;
}
```

```
//Simple HelloWorld program – ARM part
...
PROC_Setup();
PROC_Attach(ID_PROCESSOR, NULL);
PROC_Load(ID_PROCESSOR, "Helloworld_dsp_part.out", NUM_ARGS, NULL);
PROC_Start(ID_PROCESSOR);
...
```



Содержание

- Оптимизации PNX
- Texas Instruments DaVinci
- Stretch
  - Введение
  - Устройство
  - Программирование
- NVIDIA Tegra
- ARM11
- Сравнение

Stretch S6000

Введение

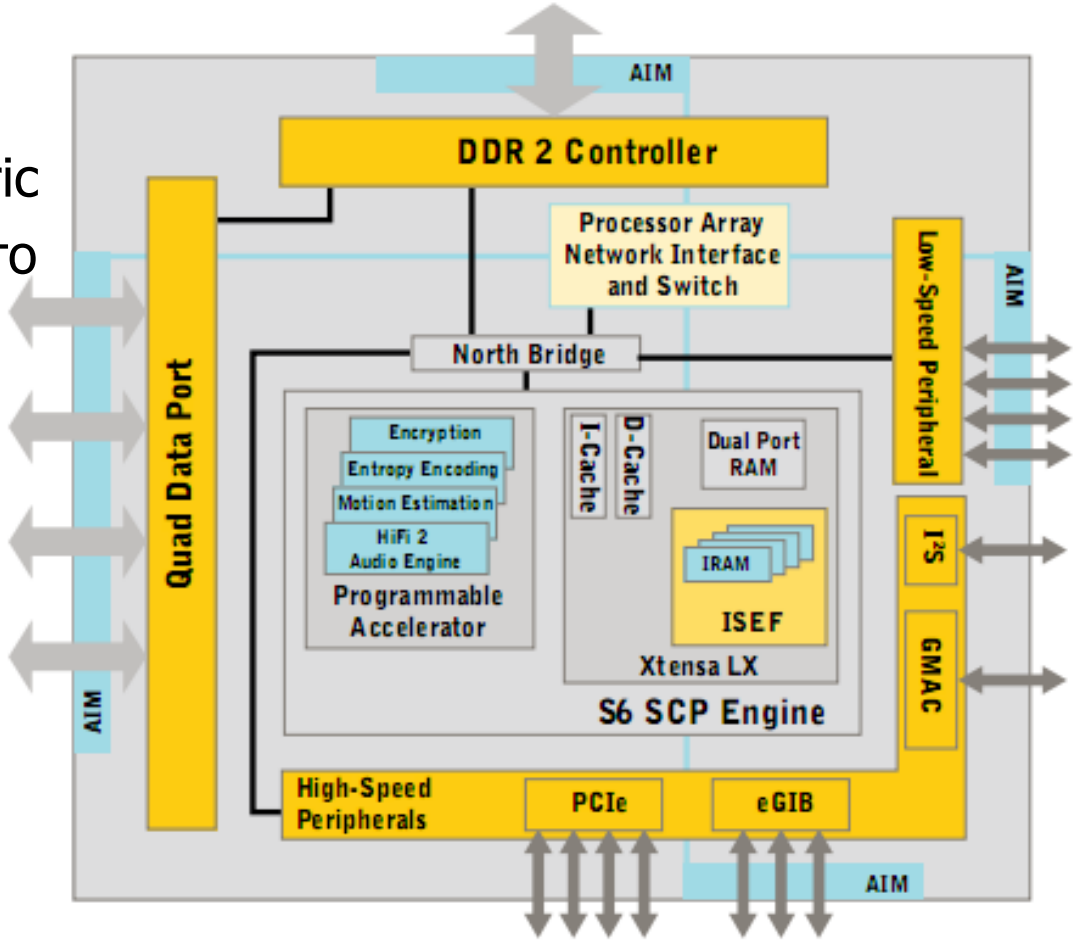
- Область применения – обработка видео
- Конфигурируемый процессор
- Возможность кластеризации
- Кодирование в реальном времени H.264 1xHD или 4xSD
- Ориентация на параллельные вычисления
- Цена – от 25$

Stretch®
Extending the Possibilities®

Stretch S6000

Устройство

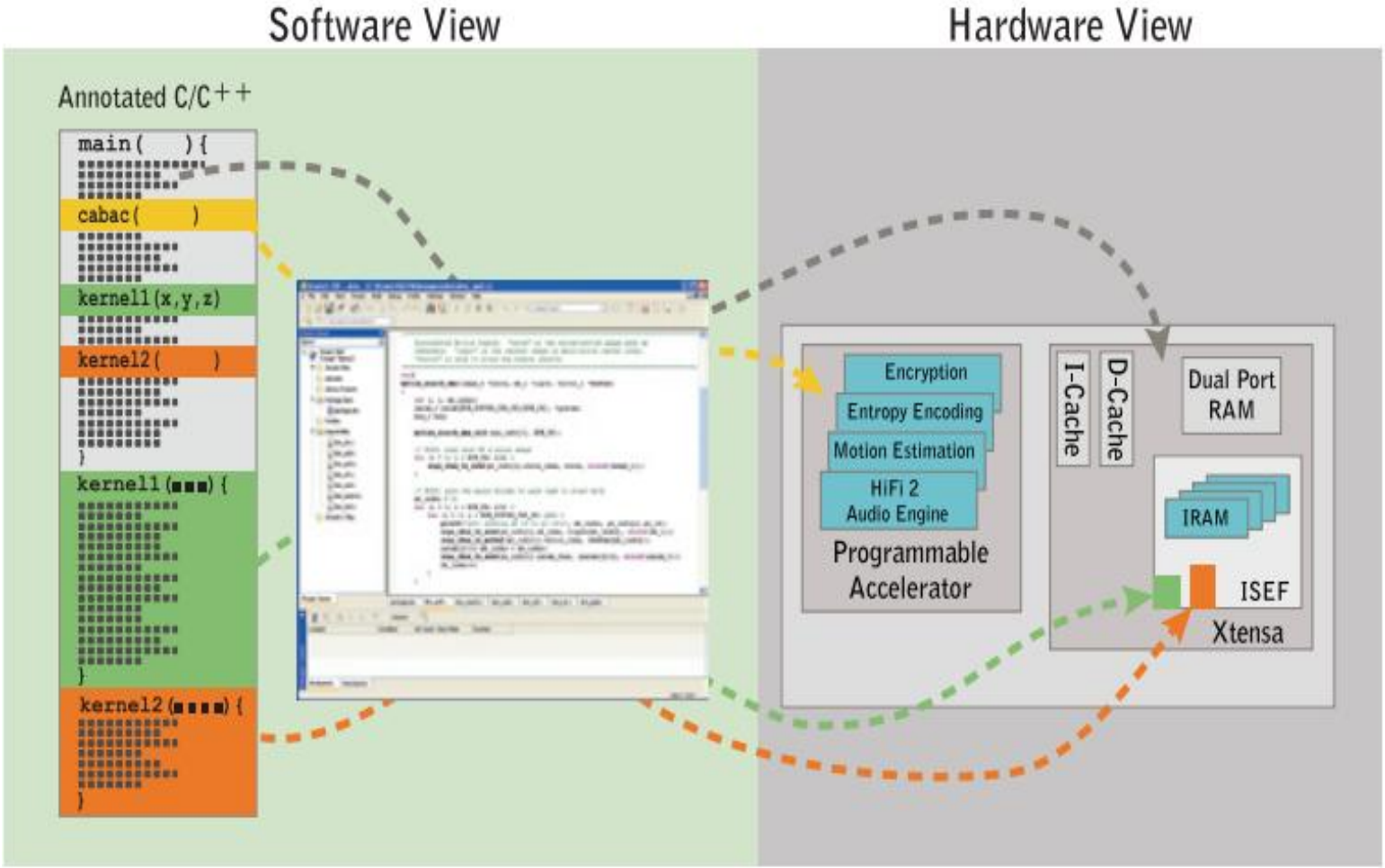
- VLIW
- Instruction Set Extension Fabric
- Аппаратная реализация часто встречающихся функций
- Оптимизирован для H.264
- 64 KB RAM
- 4096 ALU
- 32 128-битных регистра
- Изменение ISEF за 27 uS
- Ввод-вывод до 2.4 GB/S
- 167-345 MHz

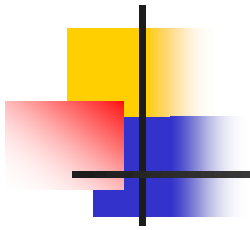


Stretch S6000

Программирование

- C/C++
- Stretch C
- Assembler





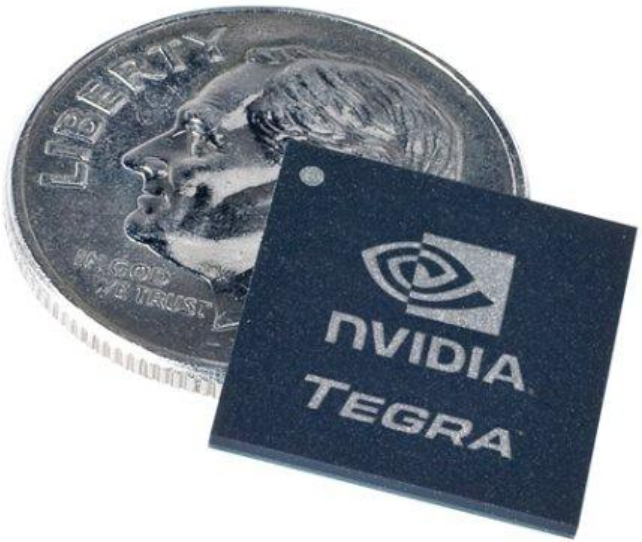
Содержание

- Оптимизации PNX
- Texas Instruments DaVinci
- Stretch
- NVIDIA Tegra
- ARM11
- Сравнение

NVIDIA Tegra

Введение

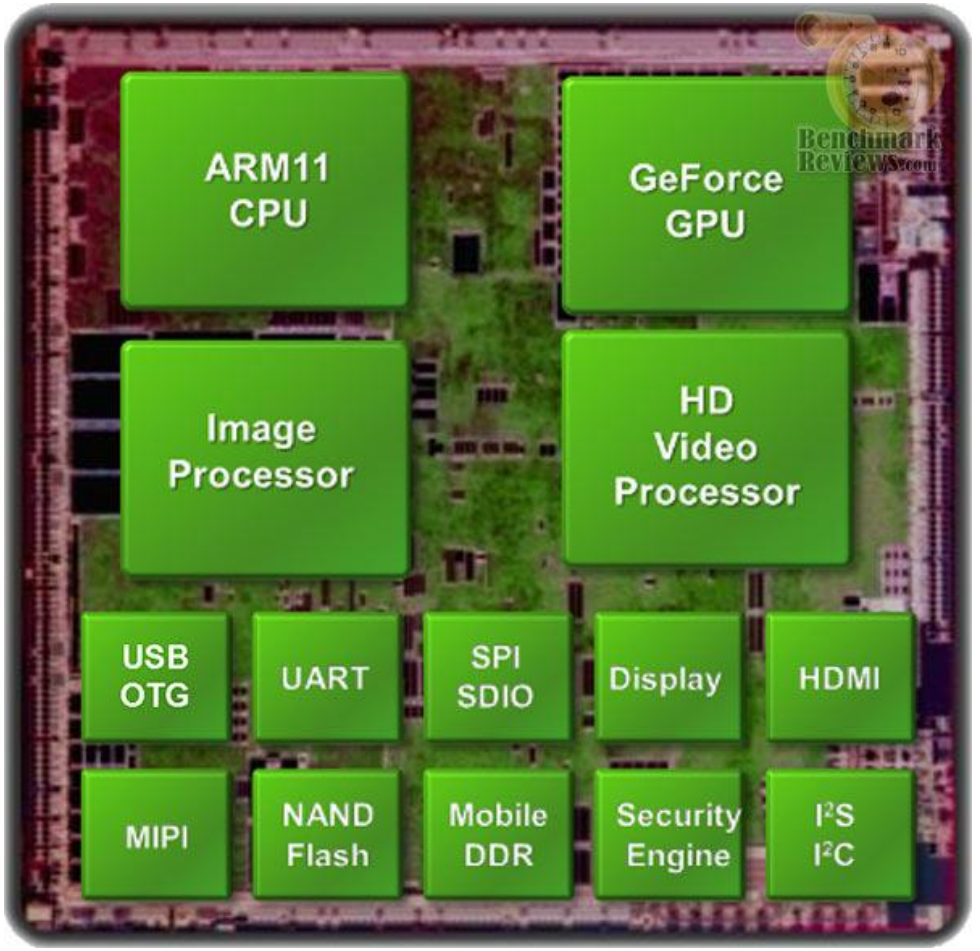
- Анонс – 2 июня 2008
- Первое коммерческое применение –сентябрь 2009 (Microsoft Zune HD)
- Область применения – мобильные устройства, GPS, смартфоны
- Ядро – ARM11, 600-900 MHz

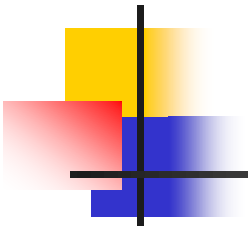


NVIDIA Tegra

Устройство NVIDIA Tegra APX 2500

- Ядро – ARM11, 600 MHz
- Графический сопроцессор ULP GeForce
- Выход на LCD до 854×480
- Поддержка камеры до 12 МП
- OpenGL ES 2.0
- Direct3D Mobile
- Декодирование H264 – 720p, 30FPS





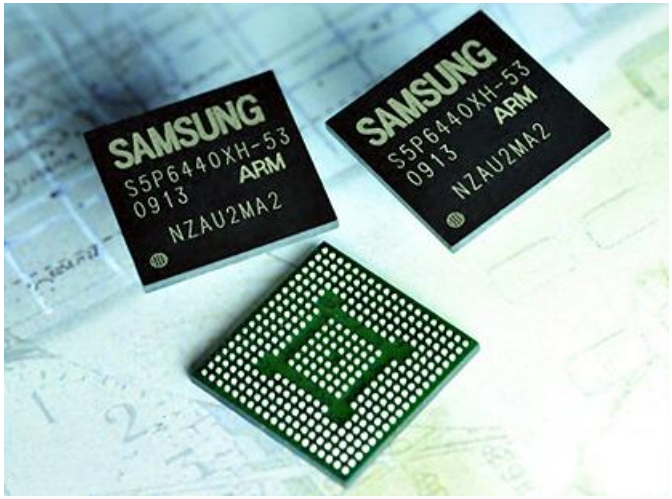
Содержание

- Оптимизации PNX
- Texas Instruments DaVinci
- Stretch
- NVIDIA Tegra
- ARM11
  - Введение
  - Устройство
  - Примеры команд
- Сравнение

ARM11

Введение

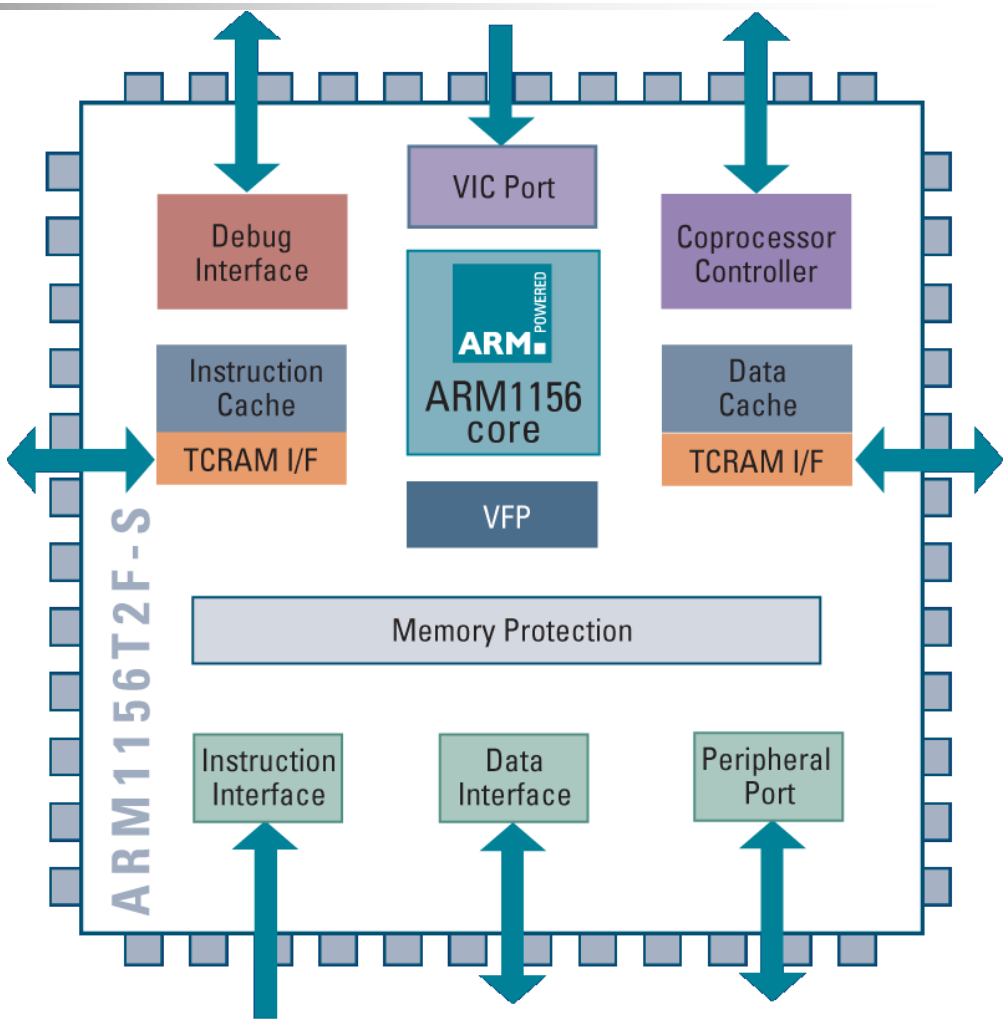
ARM



- Архитектура – RISC
- Область применения – мобильные устройства, встроенные системы
- Частота – 300-700 MHz
- Потребление - 0.6мВт/МГц
- Набор команд для DSP
- Эффективное выполнение JAVA-байткода
- Этот процессор используется в Apple iPhone!

ARM11
Устройство

- Технология – 90nm
- 32/16-битная архитектура
- Набор инструкций Thumb-2
- Набор SIMD-инструкций
- Управляемый кэш
- Расширяемость за счёт сопроцессоров
- Global Branch Prediction Unit
- Аппарат защиты памяти
- Система отладки CoreSight™
- Операции с фиксированной и плавающей точкой

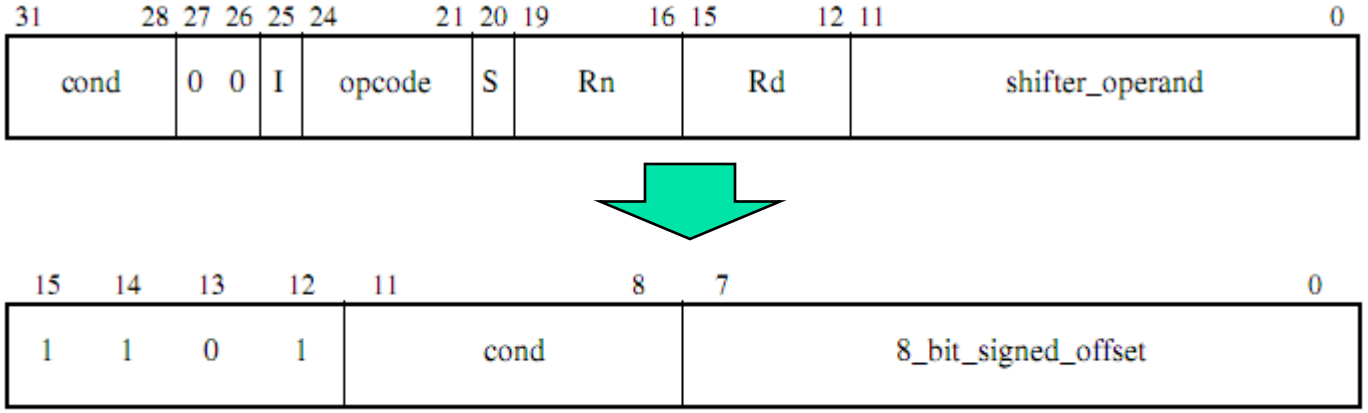


ARM1156T2-S Reference Guide
http://arm.com/products/CPUs/ARM1156T2-S.html

ARM11

Примеры инструкций

- SIMD-инструкции
  - USAD8
  - SUB16
- Thumb-инструкции



Выигрыш ~30% по скорости и компактности кода

ARM11

Примеры инструкций

- Инструкции управления кэшем
  - PLD
- Инструкции взаимодействия с сопроцессорами
  - MCR/MRC
  - LDC
- Инструкции управления GBPU
  - PrefetchFlush

ARM11

Условное выполнение команд

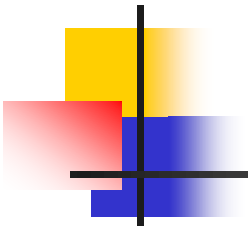
```
//исходный код на C
int gcd(int a, int b)
{
    while (a != b)
      {
        if (a > b)
            a = a - b;
        else
            b = b - a;
      }
    return a;
}
```

```
//«Прямая» реализация на ARM
gcd     CMP     r0, r1
        BEQ     end
        BLT     less
        SUB     r0, r0, r1
        B       gcd
less
        SUB     r1, r1, r0
        B       gcd
end
```

```
//«Умная» реализация на ARM
gcd
        CMP     r0, r1
        SUBGT   r0, r0, r1
        SUBLE   r1, r1, r0
        BNE     gcd
```

Реализация	Циклов на итерацию	
	Худший случай	Лучший случай
«Прямая»	17	11
«Умная»	10	10

Example of the use of conditional execution
http://www.keil.com/support/man/docs/armasm/



Содержание

- Оптимизации PNX
- Texas Instruments DaVinci
- Stretch
- NVIDIA Tegra
- ARM11
- Сравнение

Сравнение

SoC	Назначение	Область применения	Сложность программирования	Мощность	Цена
PNX	DSP	Видеонаблюдение, обработка звука	Низкая	1-4 W	30-35 $/чип 3000 $/DevKit
DaVinci	DSP, 3D-графика, CPU	Встроенные системы, видеонаблюдение	Средняя	0.5-2 W	10-90 $/чип 3-10 тыс.$/DevKit
ARM11	CPU	Мобильные устройства, встроенные системы	Низкая	0.2-1.2 W	10-30 $/чип 1000 $/DevKit
Tegra	Обработка видео, 3D-графика, Функции CPU	Мобильные устройства	?	?	?
Scratch	Обработка видео	Встроенные системы	Выше среднего	0.7-1.5 W	25-35 $/чип 3000 $/DevKit

Сравнение

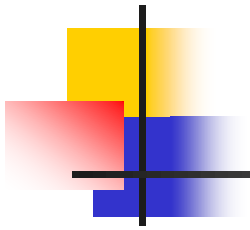
SoC	Преимущества	Недостатки
PNX	Хорошая документация, простота «первого шага», невысокая цена	Узкий модельный ряд, низкое соотношение производительность/потребляемая мощность
DaVinci	Широкая линейка продуктов, высокая производительность для широкого круга задач, аппаратная реализация часто встречающихся функций, активное сообщество разработчиков, возможна работа полноценной ОС	Высокая цена как на чипы, так и на средства разработки
ARM11	Эффективен на «нераспараллеливаемых» задачах, возможна работа полноценной ОС, расширяемость за счёт сопроцессоров	Большое количество внешних элементов
Tegra	Объединение в одном корпусе CPU и GPU, активное «продвижение» производителем	[временно?] Отсутствие в свободной продаже средств разработки, недостаток информации
Scratch	Отличная производительность на «хорошо распараллеливаемых» задачах, характерных для обработки видео	Сложность эффективного программирования

Список литературы

- Оптимизация PNX
  - TriMedia optimization guide, TCS5.1, 2006
  - TriMedia CookBook, TCS5.1, 2006
  - Nexperia PNX100x optimization slideshow, NDK 6.1, 2009
- NVIDIA Tegra
  - NVIDIA Tegra серии APX, http://www.nvidia.ru/object/product_tegra_apx_ru.html, 2009
  - NVIDIA Makes the Future, http://cnn.com, 2009
- ARM
  - ARM Architecture reference manual, http://arm.com, 2008
  - ARM DSP-Enhanced Extensions, http://arm.com, 2005
  - ARM SoC Architecture, Lung Hao-Chang, 2006
  - ARM926EJ-S™ Technical Reference Manual, http://arm.com, 2008
- Stretch
  - Stretch Goals, FPGA and Programmable Logic Journal, http://fpgajournal.com, 02.08.2005
  - S6000 Family, http://stretchinc.com, 2009
  - S6 Architecture Overview, http://stretchinc.com, 2009
  - Stretch IDE Development tool, http://stretchinc.com, 2009

Список литературы

- DaVinci
  - S-PBGA-N529 Package description, http://ti.com, 2009
  - TMS320DM644 6/3 Power Consumption Summary, http://ti.com, February 2008
  - DaVinci™ Technology Overview , http://ti.com, September 2008
  - Using the DSP in the Dual-Core DaVinci as a Graphics Render Engine, http://ti.com, March 2007
  - TMS320DM644x™ Processors–Video Benchmarks, http://ti.com, 2006
  - TMS320DM644x Digital Media Processors, http://ti.com, 2007
  - HD Video Surveillance IP Camera Reference Designs , http://ti.com, 2007
  - TMS320C6000 CPU and Instruction Set Reference Guide, http://ti.com, 2007
  - TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide, http://ti.com, October 2008
  - TMS320DM6446 DVEVM v2.0 Getting Started Guide , http://ti.com, December 2008
  - TMS320DM646x DMSoC ARM Subsystem Reference Guide, http://ti.com, September 2009
  - DaVinci™-Based Products: TMS320DM355 Processors, http://ti.com, 2008
  - Programming Details of Codec Engine for DaVinci™ Technology , http://ti.com, 2006
  - TMS320DM6446 Digital Media System-on-Chip Silicon Revisions 2.1, 1.3, 1.2, and 1.1 Silicon Errata , http://ti.com, September 2009
  - Digital Video Using DaVinci SoC, http://ti.com, 2007
  - DaVinci™ Technology Background and Specifications , http://ti.com, 2007
  - TMS320DM6446 Digital Media System-on-Chip , http://ti.com, 2007



Вопросы

?