



# 龙芯 2E 多媒体 优化简介

currentversion: 0.2

comcat <jiankemeng@gmail.com>

Sun Wah Hi-Tech System Software Ltd.

# 主要内容

---

媒体信息表示  
多媒体增强基本思想  
龙芯多媒体支持  
多媒体指令集  
优化建议



# 多媒体信息表示

# 一个颜色点的表示 ( RGB )

---

当前显示 ( 1024X768 24 位色 ) ， 需要多少存储空间？

颜色空间量化 ( 数字化 )

三原色

24 位色 ( 真彩色 ) 的数字表示



3 个字节，每个字节描述一种颜色的强度， 0xFF 最强， 0x00 最弱

0xFF0000 -----> 红色

0x0000FF -----> 蓝色

# YUV 颜色空间

---

按照亮度、色差原理来描述颜色 (Y: 亮度; U: 蓝色差; V: 红色差)

Y 与 UV 独立

JPG, MPEG 对信息的存储基于 YUV

显卡、显示器等硬件基于 RGB

JPG, MPEG 文件, 要显示出来, 会有一个转换 YUV2RGB:

$$R = Y + 1.4075 * (V - 128)$$

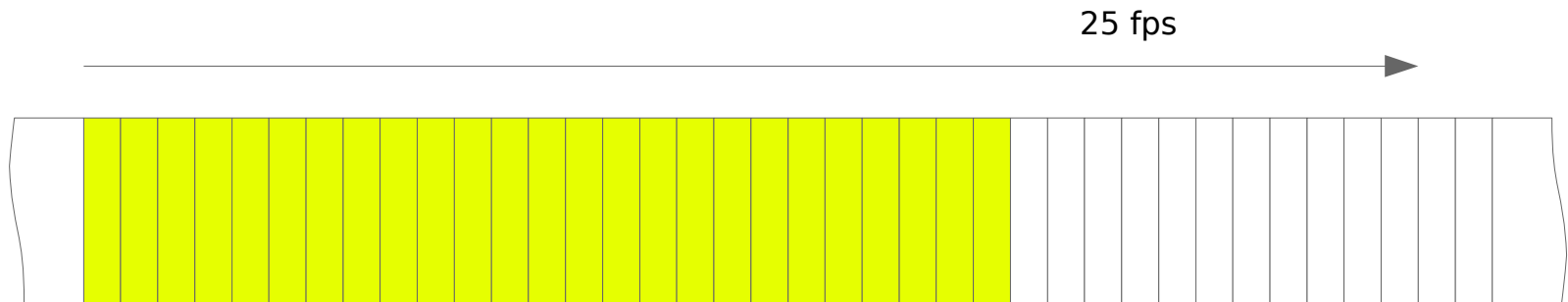
$$G = Y - 0.3455 * (U - 128) - 0.7169 * (V - 128)$$

$$B = Y + 1.779 * (U - 128)$$

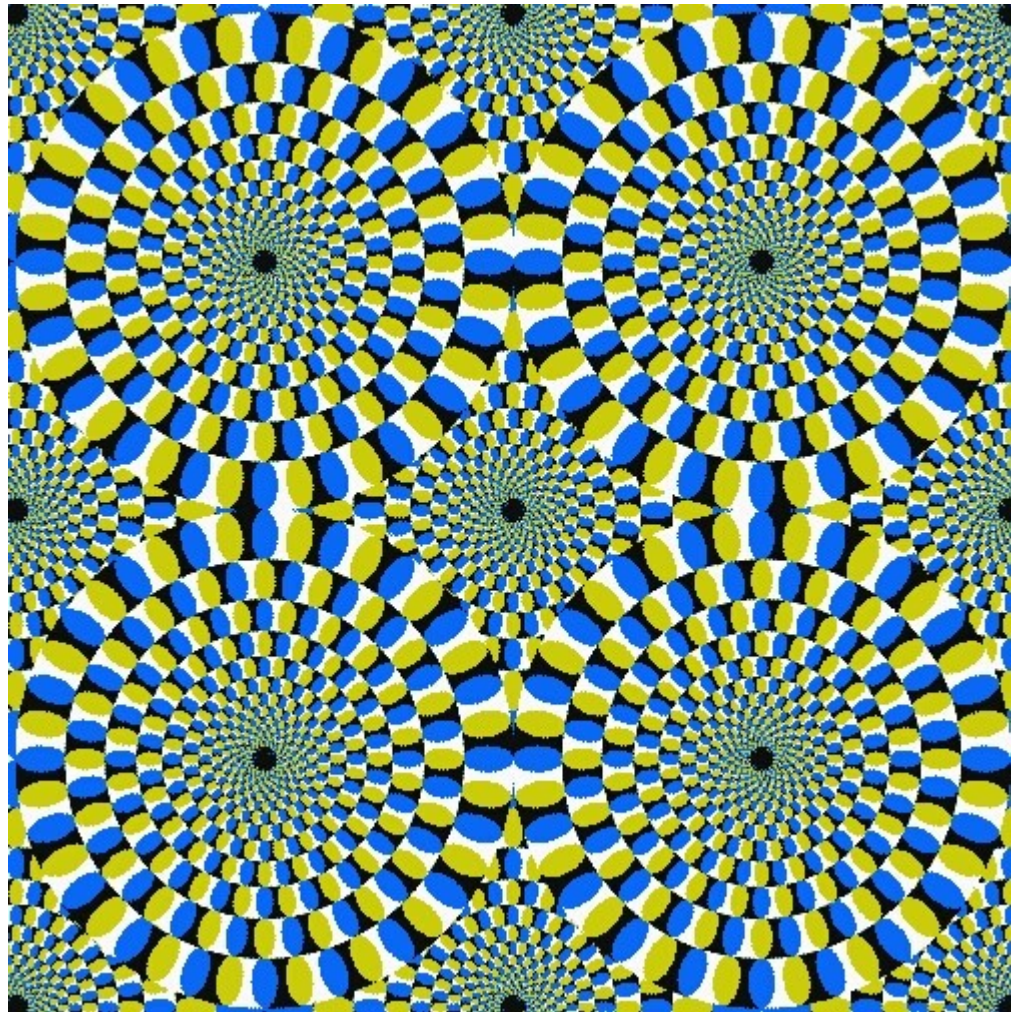
# 图像视频的表达

图像 ---> 点聚集成面 (bmp 格式)

视频 ---> 面聚集成流



以每秒 25 幅图像（帧）的速度显示，人眼“看起来”就是动画



# 多媒体增强基本思想

# 基本思想

---

典型的，以解码 MPEG 4 视频流为例，每一帧都要经过 IDCT、运动补偿、反量化、逆扫描、逆预测、变长解码等运算。一定时间内，CPU 的大部分时间都是在进行这些运算。（有限的运算类型重复）

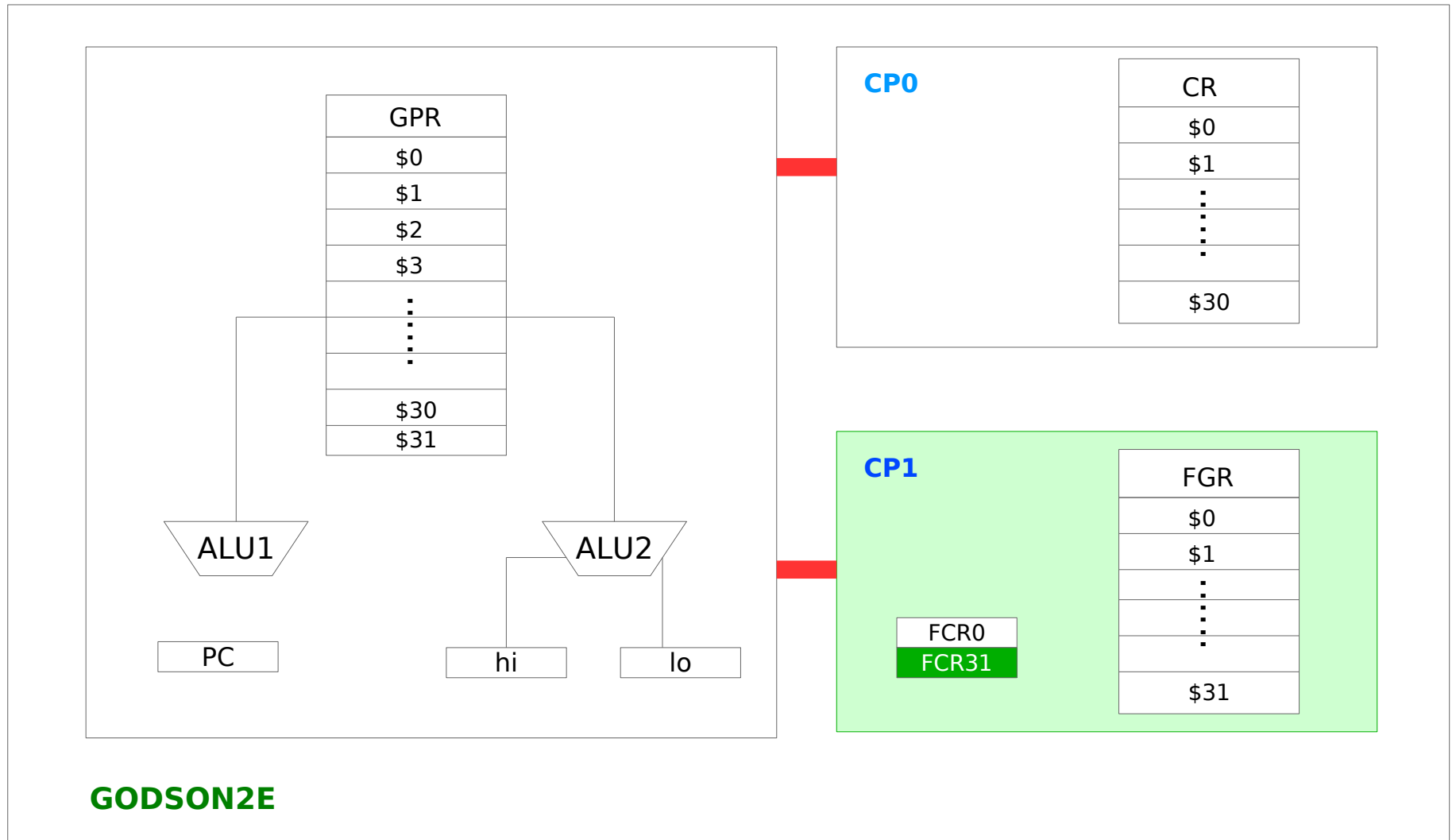
开发并行性

数据级并行性 (SIMD)，即一条指令对位于多媒体寄存器中的多个像素进行相同的操作

MMX, SSE

# 龙芯多媒体支持

# 龙芯 2E 逻辑结构



## 2E 多媒体支持

---

引入 MMX、SSE 的思想，实现了 x86 的 MMX 和部分 SSE 指令

复用浮点运算单元，定点多媒体指令由浮点协处理器 (CP1) 负责执行

浮点部件定点增强 (for, fxor, fand...)

## 2E 多媒体寄存器

使用 32 个 64 位浮点寄存器作为多媒体寄存器

默认兼容 O32，即只能使用 16 个偶数号的寄存器即：\$f0 \$f2 .. \$f30

要使用全 32 个，则将系统协处理器 (CP0) 之 12 号状态寄存器的 26 位 (FR) 置 1 即可

FCR31 号寄存器

31 - 26	23	22 - 18	17 - 12	11 - 7	6 - 2	1-0
---------	----	---------	---------	--------	-------	-----

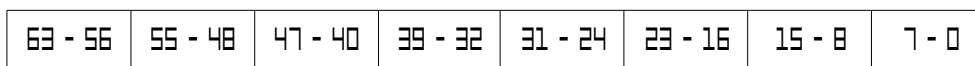
FCR31 格式

# 多媒体指令概述

## 2E 多媒体指令概述

- 龙芯 2E 多媒体指令的操作对象为 64 位，可以分割成  $8*8$ ,  $16*4$ ,  $32*2$ ,  $64*1$ 。如果每个单元为 8 位（称为 Packed byte）则有 8 个这样的单元；若每个单元为 32 位（Packed word）则有 2 个这样的单元。而每条指令可以同时对这这几个单元进行操作（SIMD）

Packed Byte



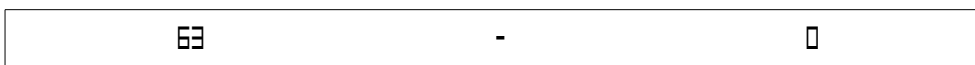
Packed Half-word



Packed Word



Packed Doubleword



# 优化建议

# 优化建议

---

借鉴 x86 的成果，避免重复工作，浪费资源

算法层面上，优化的基本思想是寻找算法的并行性，即分拆大法  
(对一个单元的一连串计算，分拆成对多个单元的 MMI 描述)

如果可能的话，可以从系统结构层面上考虑问题，比如提高  
cache 的命中率、使用寄存器时避免数据冒险、短循环展开避免  
控制指令等等。

谢谢!

