FACULTY
OF INFORMATICS
Masaryk University

Curso de Desenvolvimento GBA

1. Introdução ao GBA

João Paulo Taylor Ienczak Zanette

Índice

Objetivos

- Ensinar programação de baixo-nível (comunicação direta com hardware/integração com assembly);
- Ensinar técnicas de programação aplicadas;
- Mostrar o funcionamento de imagens/gráficos e áudio no mundo digital;
- Relacionar as tecnologias vistas com as utilizadas atualmente.

Programação

- Assembly ARM7-TDMI — Modo Thumb (GBA);
- OpenGL (NDS);
- C++ (GBA/NDS).

A mesma forma de programação para GBA serve também para: GB, GBC, NES, SNES, MegaDrive, SegaSaturn e PSX (PS1).

- Leitura/escrita de registradores (em que cada bit é mapeado para uma função específica) via programação;

Sistemas Digitais

- Compreensão a respeito de como o Assembly gerado pela compilação altera o estado/memória do circuito;
- Compreensão do sistema que gera imagens em um circuito digital (VGA, LCD, etc...);
- Funcionamento (inclusive a nível de circuito) da execução de músicas em formato de instrução MIDI;
- Técnicas de otimização através de Hardware.

Computação Gráfica

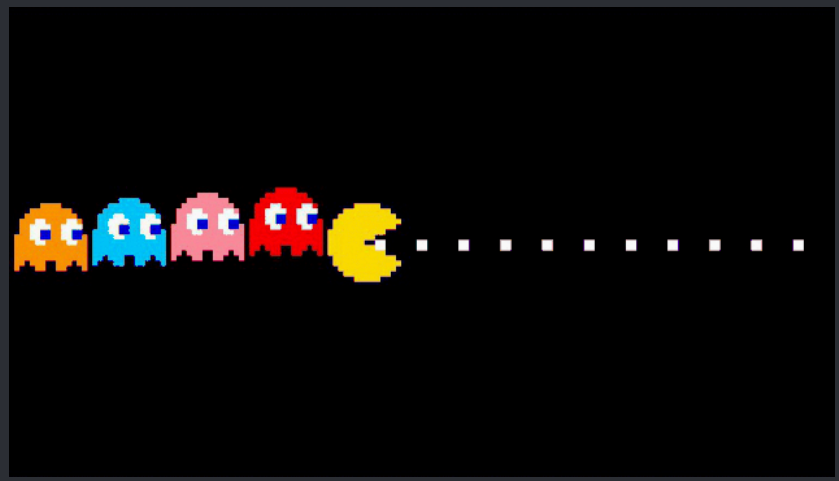
- Desenho de primitivas (linhas, triângulos, circuitos, etc...);
- Aceleração gráfica via Hardware.

Notações

`0b???`	«`???`» está em binário (`0b11111111 = 255`).
`0x???`	«`???`» está em hexadecimal (`0xFF = 255`).
`???b`	«`???`» está em binário (`11111111b = 255`).
`???h`	«`???`» está em hexadecimal (`FFh = 255`).
`????:????`	«`:`» serve apenas para melhor visualização (`0x04000000 = 0400:0000h`).
`??? >> x`	«`???`» deslocado “x” bits para direita.
`??? << x`	«`???`» deslocado “x” bits para esquerda.

Hello, World!

Um breve prólogo das especificações técnicas e limitações dos consoles de VideoGame ao longo do tempo e uma análise de sua evolução.



Atari 2600 (1977)

Processador: MOS Technology 6507 (variante do 6502 de 1975)

Barramento: 8 bits

Clock: 1.19MHz

RAM: 128 bytes

ROM: 16KB

Resolução:

- 160×192 (NTSC)
- 160×228 (PAL)

Cores: 128

Som: 2 canais (1 chip cada)

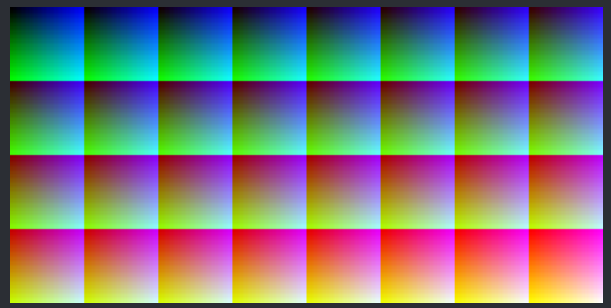
NES (1983)

Processador: MOS 6502 Customizado
Barramento: 8 bits
Clock (CPU): 1.79MHz (NTSC), 1.66MHz (PAL)
Clock (GPU): 5.37MHz (NTSC), 5.33MHz (PAL)
RAM: 2KiB + RAM Expandida (do cartucho)
ROM: 48KB
Resolução: 256×240
Cores: 56 cores (paleta básica)
Cores na tela: 25 cores por scanline (cor de fundo + 4 conjuntos de 3 cores de tiles + 4 conjuntos de cores por sprite)
OAM: 256 bytes
Dim. das Sprites: 16×16 ou 24×24
Máx. Sprites na tela: 64
Som: 5 canais (2 square, 1 triangle, 1 ruído-branco, 1 modulação de código delta-pulse (DPCM) de 6 bits)

SNES (1990)

Processador: Ricoh 5A22 customizado da Nintendo
Barramento: 16 bits
Clock (CPU): 1.79MHz, 2.86MHz ou 3.58MHz
Clock (GPU): Mesmo da CPU
RAM: 128KB
VRAM: 64KB (512 + 32 bytes de sprite, 256×15 bits de paleta)
RAM (Áudio): 64KB

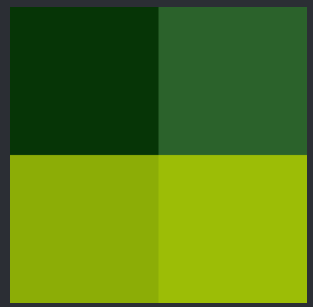
Resolução: 256×224/512×448
Cores: 32768 (15 bits)
OAM: 544 bytes
Dim. das sprites: 8×8, 16×16, 32×32 e 64×64
Cores/sprite: 16
Máx. sprites na tela: 128 (32 na mesma linha)
Camadas de background: 4
Som: 8 canais (32KHz 16-bit stereo).



GameBoy (1989)

Processador: Sharp LR3590Z Customizado
Barramento: 8 bits
Clock (CPU): 4.19MHz
RAM: 8KB (podendo ser expandido para 32KB)
ROM: 256 bytes (interno), 256K/512K/1M/2M/4M/8M (cartuchos)

VRAM: 8KB (interno)
Resolução: 160x144
Cores: 2 bits (4 tons de «cinza»)
OAM: 160 bytes (4 bytes/sprite)
Dim. das sprites: 8x8, 8x16
Cores/sprite: 16
Máx. sprites na tela: 40
Som: 2 geradores de pulso de onda



GameBoy Advance (2001)

Processador: ARM7-TDMI com memória embarcada

Barramento: 16 bits

Co-processador: Z80 8-bit de 4/8MHz (para compatibilidade com GB/GBC)

Clock (CPU/GPU): 16.8MHz/~5.5MHz (**59.73FPS**)

SRAM/DRAM: 32KB/256KB

VRAM: 92KB (interno)

Resolução: 240x160 (3:2)

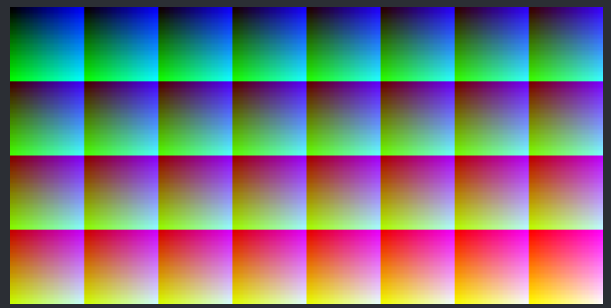
Cores: 15-bit BGR (5 bits/canal), 512 cores (character mode), 32768 cores (bitmap mode)

OAM: 1KB (128 objetos de 3x16bit, 32 transformações de objetos de 4x16bit)

Dim. das sprites: 8x8, 8x16, 8x32, 16x8, 16x16, 16x32, 32x8, 32x16, 32x32, 32x64, 64x32, 64x64

Máx. sprites na tela: 256

Som: Dual 8-bit DAC para som Stereo (DirectSound)



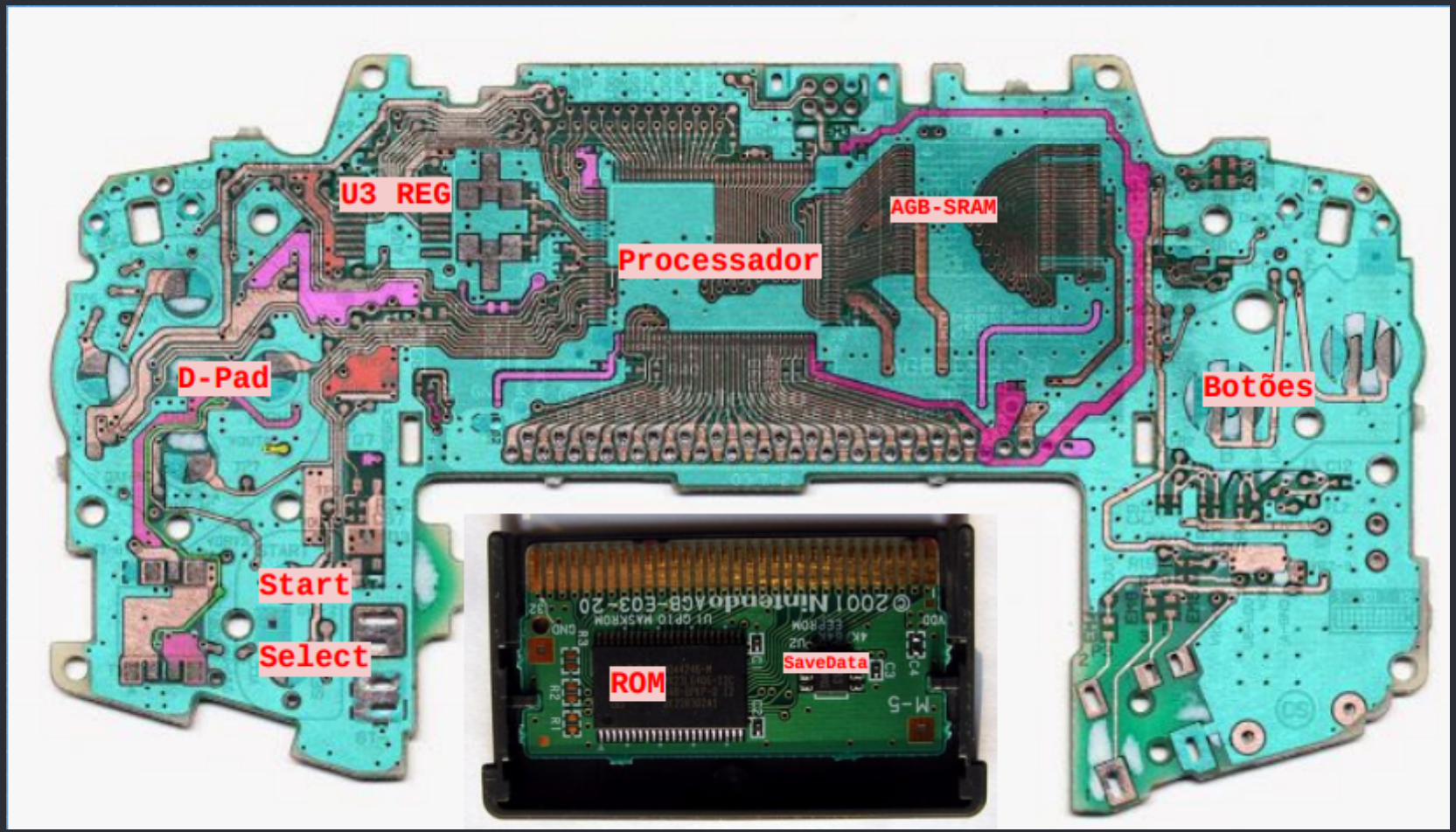
GameBoy Advance (2001)

ARM7-TDMI : ARM7 + 16-bit Thumb + JTAG Debug + fast Multiplier + enhanced ICE.

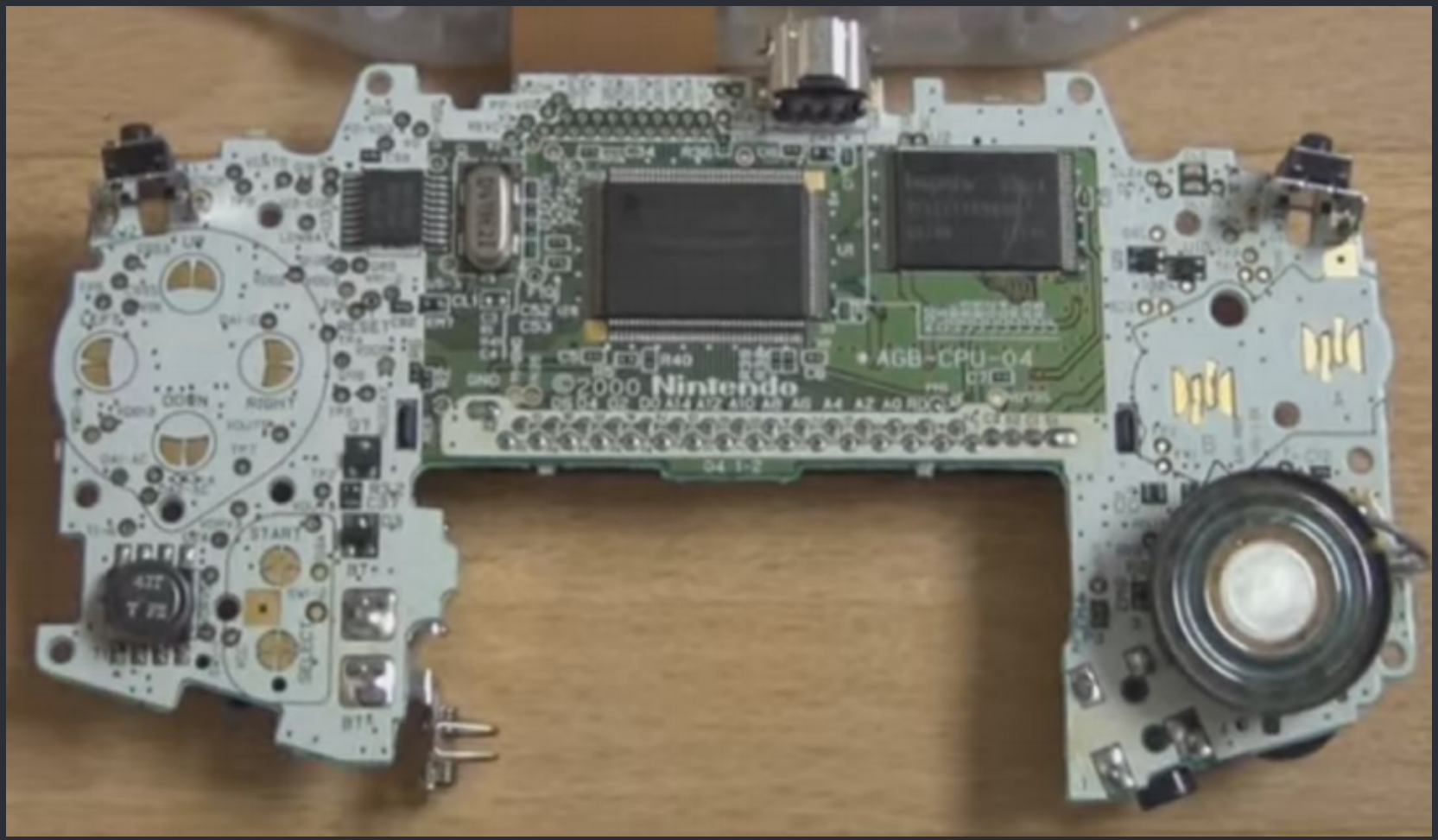
DAC: Digital-Analogic-Converter

O interior do GBA

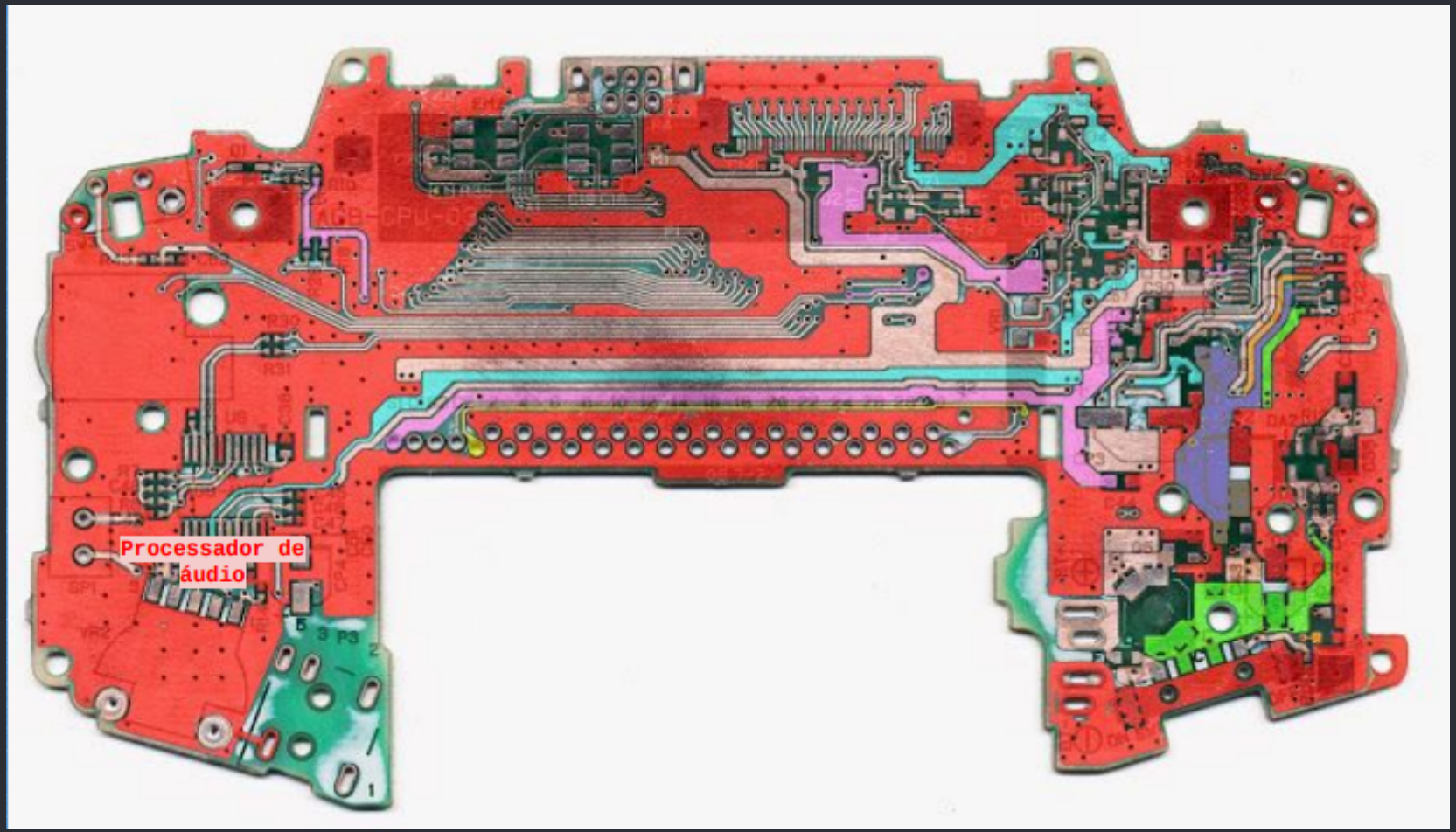
Interior do GBA (frontal)



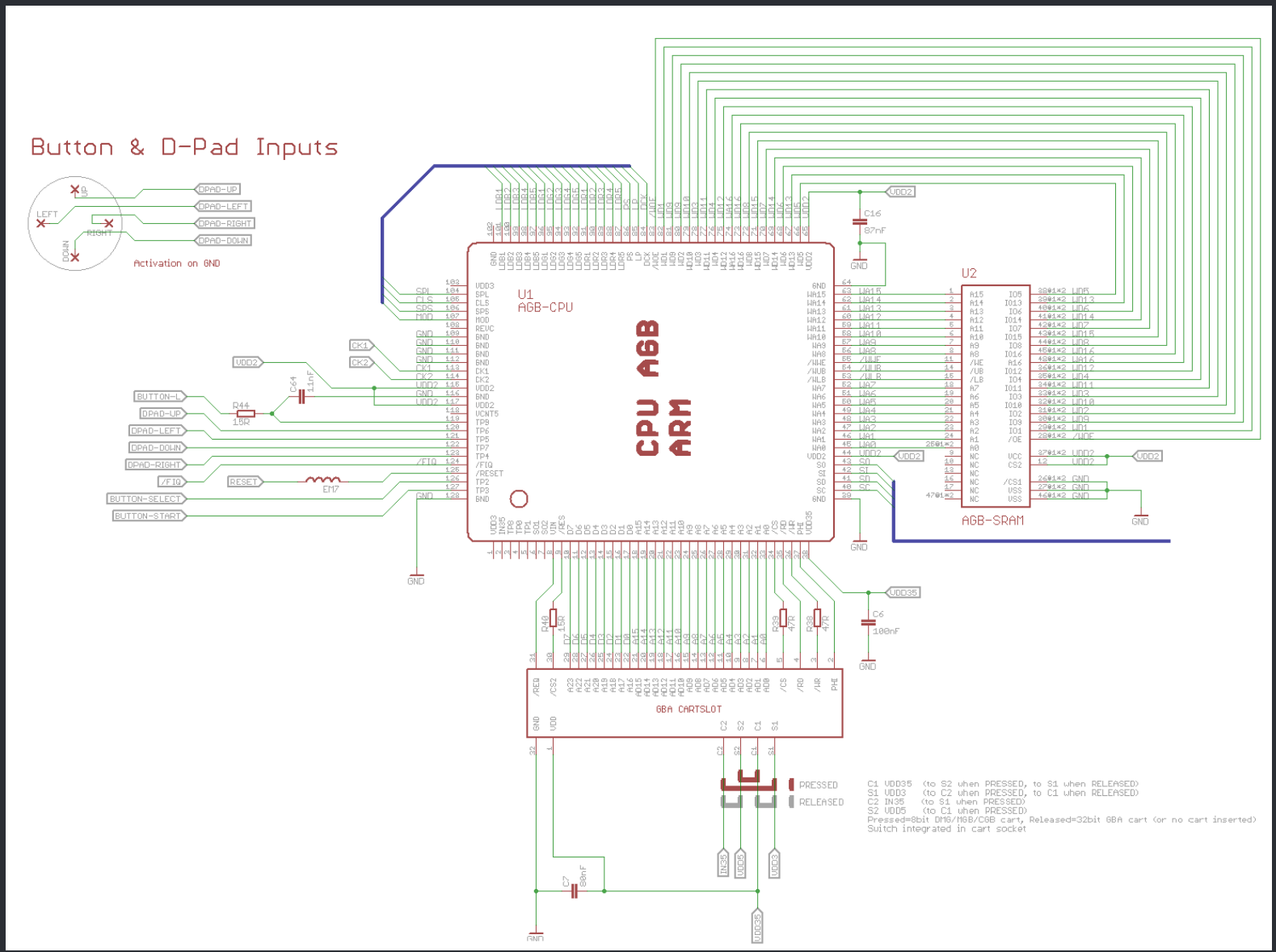
Interior do GBA (frontal)



Interior do GBA (trás)



Interior do GBA: Processador



Organização da Memória (Geral)

Descrição	Início	Tamanho
BIOS – ROM do Sistema	`0x00XX:XXXX`	`0x0:3FFF (16KB)`
WorkRAM On-Board	`0x02XX:XXXX`	`0x3:FFFF (256KB)`
WorkRAM On-Chip	`0x03XX:XXXX`	`0x0:7FFF (32KB)`
Registradores de I/O	`0x04XX:XXXX`	`0x0:03FE (~1KB)`

Organização da Memória (Vídeo)

Descrição	Início	Tamanho
Paleta de BG/OBJ	`0x05XX:XXXX`	`0x0:03FF (1KB)`
RAM de Vídeo (VRAM)	`0x06XX:XXXX`	`0x1:7FFF (96KB)`
OAM (Obj. Attr. Mem.)	`0x07XX:XXXX`	`0x0:03FF (1KB)`

Organização da Memória (GamePak)

Descrição	Início	Tamanho
FlashROM (ws0)	`0x08XX:XXXX`	`0x1FF:FFFF (16KB)`
FlashROM (ws1)	`0x0AXX:XXXX`	`0x1FF:FFFF (256KB)`
FlashROM (ws2)	`0x0CXX:XXXX`	`0x1FF:FFFF (32KB)`
GamePakSRAM	`0x0EXX:XXXX`	`0x000:FFFF (64KB)`

Vídeo

Vídeo: Funcionamento

Vimos que o GBA possui uma tela com resolução 240x160 pixels. Porém, isso varia do ***modo de vídeo*** (Video Mode).
No GBA há 3 modos ***Bitmapeados*** (enumerados como 3, 4 e 5) e 3 modos ***Tilemapeados*** (enumerados como 0, 1 e 2). Por simplicidade, começaremos com modos Bitmapeados (especificamente o 3).

Modo	BGs	Resolução	bpp	RAM	Page-Flip
3	2	`240x160`	16	`(1x) 12C00h`	Não
4	2	`240x160`	8	`(2x) 9600h`	Sim
5	2	`160x128`	16	`(2x) A000h`	Sim

Vídeo: Registrador de Controle do Visor

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
W	W	W	OB	B3	B2	B1	B0	F	O	H	P	C	Mode		

Bit	Descrição
M (0-2)	Display-Mode.
C (3)	(Read-Only) Indica se o cartucho inserido é de GBC (1) ou GBA (0).
P (4)	Seleção de página.
H (5)	Habilita acesso à OAM quando em HBlank.
O (6)	Modo de mapeamento de objetos. 0 = 2D/Matricial; 1 = 1D/Sequencial.
F (7)	Força uma tela em branco.
B<X> (8-B)	Habilita Background <X>.
OB (C)	Habilita camada de objetos.
W (D-F)	Habilita o uso das janelas 0/1/de objetos, respectivamente. Janelas podem ser usadas como máscaras (como foi feito com o lampião em Zelda).

Vídeo: Registrador de Controle do Visor

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
W	W	W	OB	B3	B2	B1	B0	F	O	H	P	C	Mode		

Bit	Descrição
M (0-2)	Display-Mode.
C (3)	(Read-Only) Indica se o cartucho inserido é de GBC (1) ou GBA (0).
P (4)	Seleção de página.
H (5)	Habilita acesso à OAM quando em HBlank.
O (6)	Modo de mapeamento de objetos. 0 = 2D/Matricial; 1 = 1D/Sequencial.
F (7)	Força uma tela em branco.
B<X> (8-B)	Habilita Background <X>.
OB (C)	Habilita camada de objetos.
W (D-F)	Habilita o uso das janelas 0/1/de objetos, respectivamente. Janelas podem ser usadas como máscaras (como foi feito com o lampião em Zelda).

Vídeo: Registrador de Controle do Visor

O ***Mode 3*** usa o BG2 para renderizar, então precisamos habilitar o bit do Mode 3. É necessário também habilitar o próprio ***Mode 3*** em si. Para isso, o campo representado pelos *bits* 0, 1 e 2 precisa conter o valor "3", que em binário é representado por `0b11`.

…	B	A	9	8	7	6	5	4	3	2	1	0
…	B3	B2	B1	B0	F	O	H	P	C	Mode		
…	0	1	0	0	0	0	0	0	0	0	1	1
…	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

Vídeo: Registrador de Controle do Visor

…	B	A	9	8	7	6	5	4	3	2	1	0
…	B3	B2	B1	B0	F	O	H	P	C	Mode		
…	0	1	0	0	0	0	0	0	0	0	1	1
…	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

Com isso, temos:

$$2^{10} + 2^{1} + 2^{0} = 1024 + 2 + 1 \quad (1)$$

$$1027 = 0x403 \quad (2)$$

Representando a alteração do registrador de controle do visor em pseudo-código:

```
memory[0x4000000] = 0x403;
```