
Nintendo
GAME BOY
COMPACT VIDEO GAME SYSTEM
Nintendo GAME BOY TM
SELECT
START
B
OFFICIAL
GAME BOY
COMPACT VIDEO GAME SYSTEM
Is That a Demo in Your Pocket?
Code Snorpung
Music Nordloef
Original Nintendo Seal of Quality


Nintendo
GAME BOY
COMPACT VIDEO GAME SYSTEM
Nintendo GAME BOY™
SELECT
START
B
OFFICIAL
GAME BOY
COMPACT VIDEO
GAME SYSTEM
Is That a Demo in
Your Pocket?
Code Snorpung
Music Nordloef
Original
Nintendo
Seal of
Quality

33C3

2016-12-30

Michael Steil

The Ultimate Game Boy Talk

@pagetable
http://www.pagetable.com/

33C3

2016-12-30

Michael Steil

The Ultimate Game Boy Talk

@pagetable
http://www.pagetable.com/

The Ultimate Talk Series

33C3

2016

Michael Steil

The Ultimate Game Boy Talk

The Ultimate Talk Series

25C3 2008

Michael Steil

The Ultimate Commodore 64 Talk

33C3 2016

Michael Steil

The Ultimate Game Boy Talk

The Ultimate Talk Series

25C3 2008

Michael Steil

The Ultimate Commodore 64 Talk

26C3 2009

Sven Oliver ('SvOlli') Moll

The Ultimate Atari 2600 Talk

33C3 2016

Michael Steil

The Ultimate Game Boy Talk

The Ultimate Talk Series

25C3 2008

Michael Steil

The Ultimate Commodore 64 Talk

26C3 2009

Sven Oliver ('SvOlli') Moll

The Ultimate Atari 2600 Talk

29C3 2012

Tomaž Šolc

The Ultimate Galaksija Talk

33C3 2016

Michael Steil

The Ultimate Game Boy Talk

The Ultimate Talk Series

25C3 2008

Michael Steil

The Ultimate Commodore 64 Talk

26C3 2009

Sven Oliver ('SvOlli') Moll

The Ultimate Atari 2600 Talk

29C3 2012

Tomaž Šolc

The Ultimate Galaksija Talk

32C3 2015

rahra

The Ultimate Amiga 500 Talk

33C3 2016

Michael Steil

The Ultimate Game Boy Talk

DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY™
B
A
SELECT
START

118 M Game Boy &
Game Boy Color

118 M Game Boy &
Game Boy Color

79 M Game Boy Advance

118 M Game Boy &
Game Boy Color

79 M Game Boy Advance

---

197 M total

118 M Game Boy &
Game Boy Color
79 M Game Boy Advance

197 M total

1049 Game Boy Games
576 Game Boy Color Games

118 M Game Boy &
Game Boy Color
79 M Game Boy Advance

---

197 M total

1989 - 2003 Game Boy &
Game Boy Color

1049 Game Boy Games
576 Game Boy Color Games

118 M Game Boy &
Game Boy Color
79 M Game Boy Advance

---

197 M total

1049 Game Boy Games
576 Game Boy Color Games

1989 - 2003 Game Boy &
Game Boy Color
1989 - 2009 including
Game Boy Advance

Nintendo **Game Boy**

Apr 1989

Wikimedia Commons

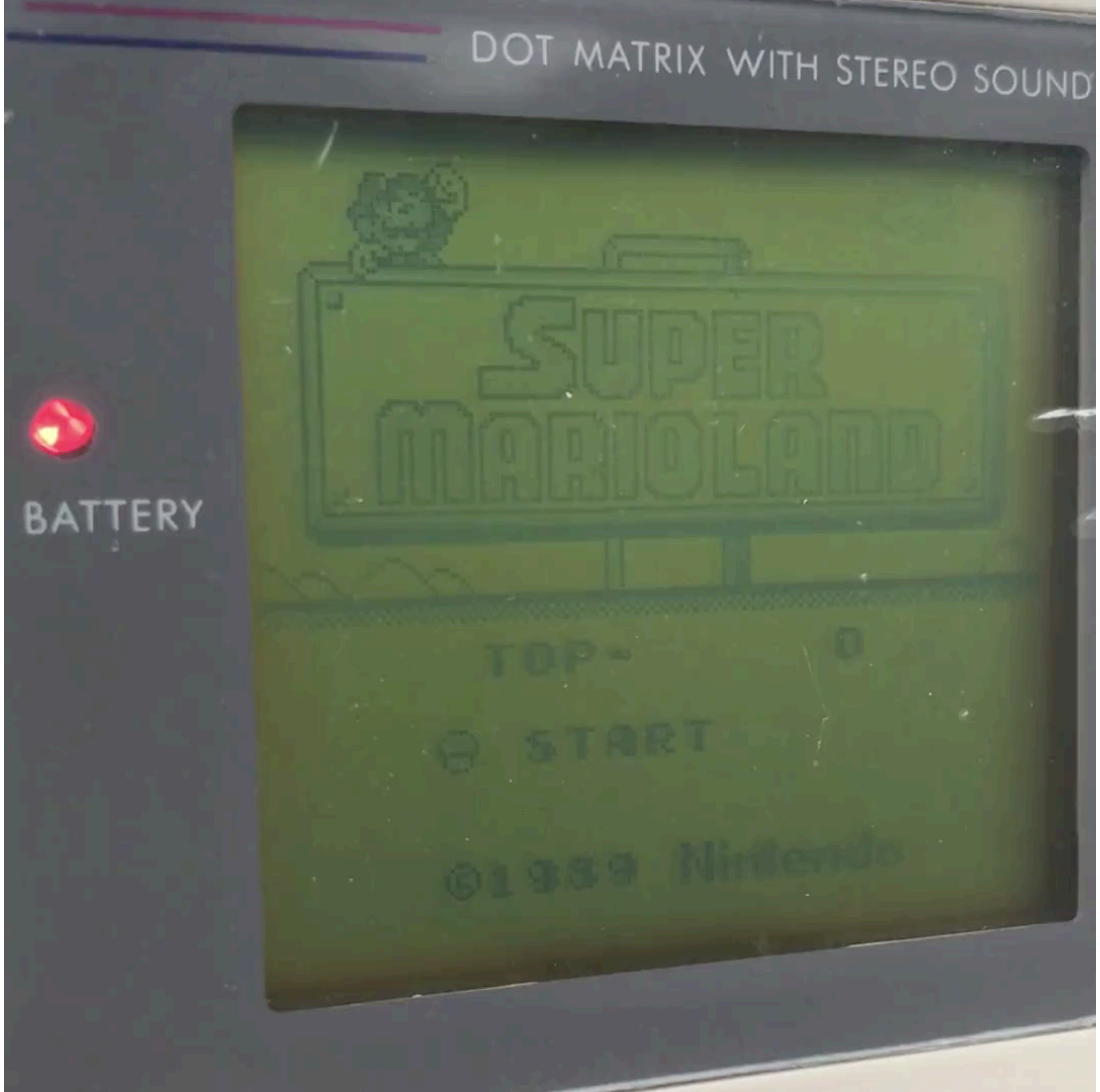
Nintendo **Game Boy**
Apr 1989

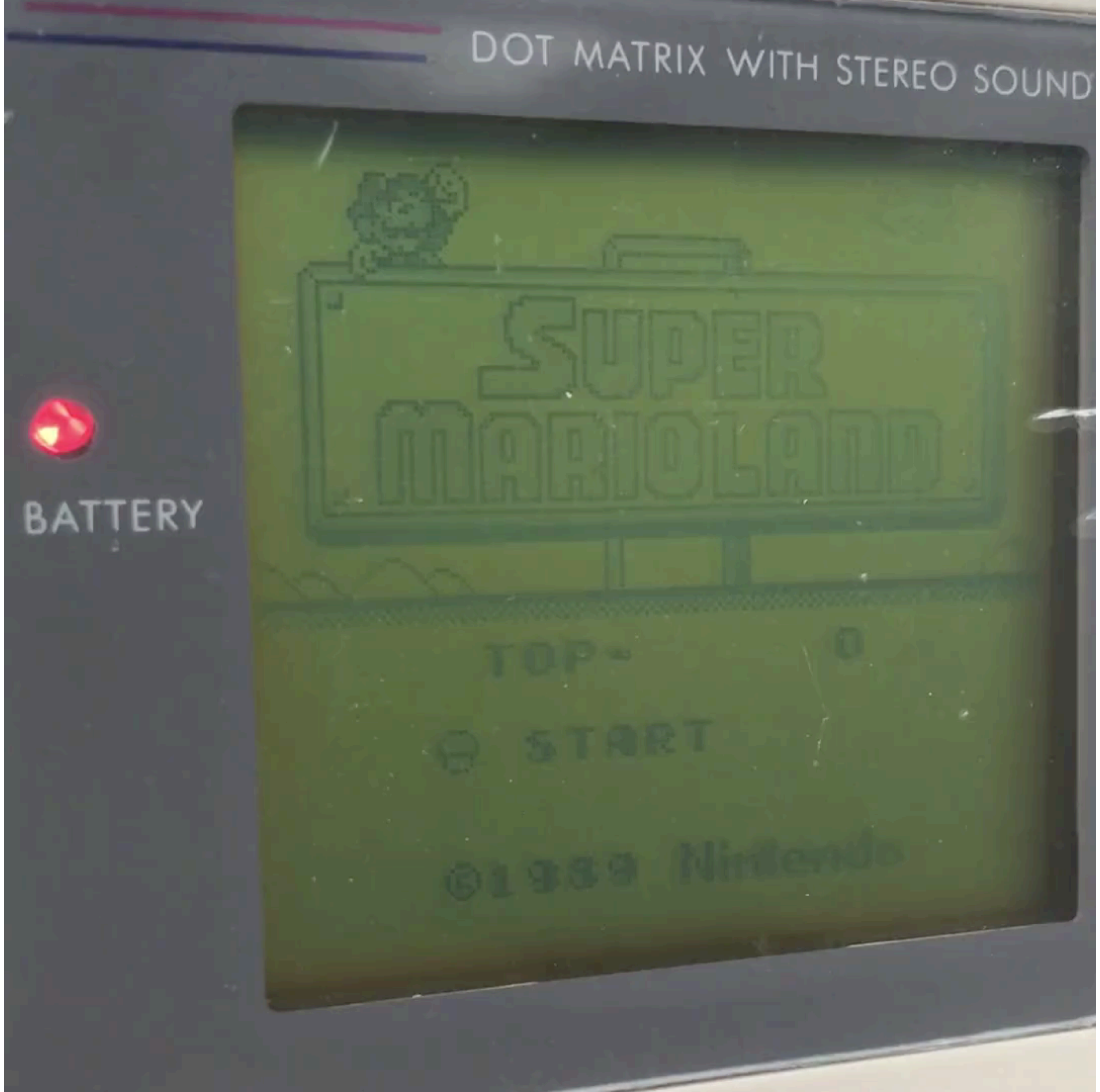
Atari **Lynx**
Sep 1989

SEGA **Game Gear**
Oct 1990

NEC **TurboExpress**
Dec 1990

Wikimedia Commons


DOT MATRIX WITH STEREO SOUND
BATTERY
SUPER
MARIOLAND
TOP- 0
START
©1989 Nintendo


DOT MATRIX WITH STEREO SOUND
BATTERY
SUPER
MARIOLAND
TOP- 0
START
©1989 Nintendo

Game Boy

DMG-01

1989

Game Boy Pocket

MGB-001

1996

Game Boy Light

MGB-101

1998

Game Boy Color

CGB-001

1998

Game Boy Advance
AGB-001

2001

Game Boy Advance SP
AGS-001

2003

Game Boy Advance SP
AGS-101

2005

Game Boy Advance

AGB-001

2001

Game Boy Advance SP

AGS-001

2003

Game Boy Advance SP

AGS-101

2005

Super
Game Boy
SHVC-027
SGB

1994

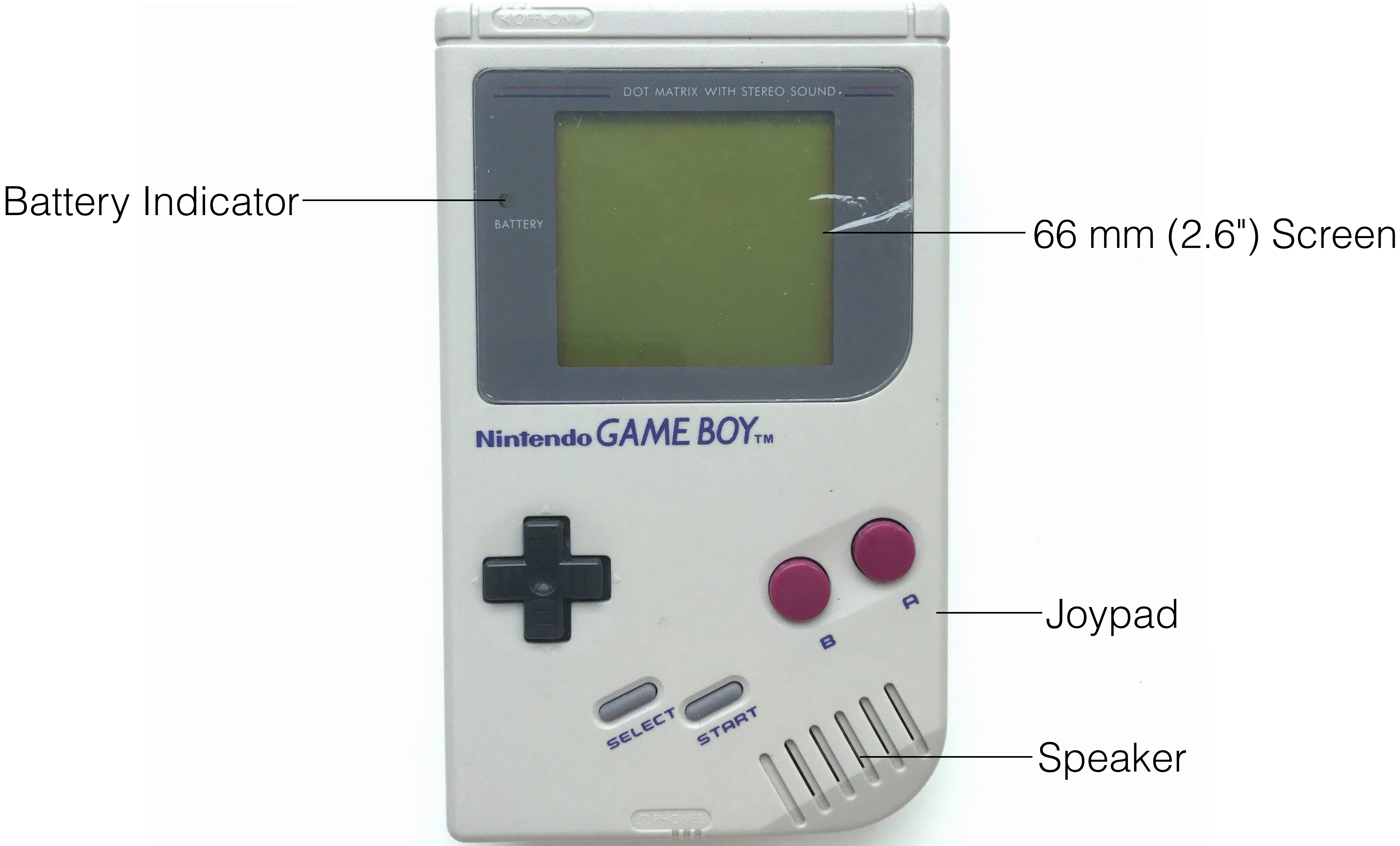
Super
Game Boy 2
SHVC-042
SGB2

1998

Game Boy
Player
DOL-017

2003

OFF•ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES


Battery Indicator
66 mm (2.6") Screen
Joypad
Speaker
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
SELECT
START
A
B


DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY™
B
A
SELECT
START

On Switch
Game Pak
Link Connector
External Power
Volume
Contrast
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY™
SELECT
START
B
A
Headphones


OFF•ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY™
B
A
SELECT
START
PHONES


G13615817
MODEL NO. DMG-01
© 1989 Nintendo
RATING : DC6V 0.7W
BATTERY : SUM-3(AA.R6)X4
PAT. PEND. MADE IN JAPAN
Complies with the limits for a Class B computing device pursuant to Subpart J of Part 15 of FCC Rules.
INSTALLATION, MAINTENANCE OR SERVICE
1-800-255-3700
NINTENDO WORLD CLASS SERVICE. CENTER HOTLINE


Game Pak
Nintendo GAME BOY
G13615817
MODEL NO. DMG-01
©1989 Nintendo
RATING: DC6V 0.7W
BATTERY: SUM-3(AA.R6)X4
PAT. PEND. MADE IN JAPAN
Complies with the limits for a Class B computing device pursuant to Subpart J of Part15 of FCCRules.
INSTALLATION, MAINTENANCE OR SERVICE
1-800-255-3700
NINTENDO WORLD CLASS SERVICE CENTER HOTLINE


Nintendo GAME BOY TM
TETRIS
G13615817
MODEL NO. DMG-01
© 1989 Nintendo
RATING: DC6V 0.7W
BATTERY: SUM-3(AA.R6)×4
PAT. PEND. MADE IN JAPAN
TÜV
VCI
Complies with the limits for a Class B computing device pursuant to Subpart J of Part 15 of FCC Rules.

Four AA Batteries


Nintendo GAME BOY TM
TETRIS
DMG-TR-USA-1
MADE IN JAPAN
THIS SIDE OUT
OFFICIAL
GAME LINK
GAME PAK
Nintendo

	Game Boy	NES	Commodore 64	Atari 2600	SNES

	Game Boy	NES	Commodore 64	Atari 2600	SNES
CPU	1 MHz* 8 bit	1.8 MHz 8 bit	1 MHz 8 bit	1.2 MHz 8 bit	3.6 MHz 16 bit

	Game Boy	NES	Commodore 64	Atari 2600	SNES
CPU	1 MHz* 8 bit	1.8 MHz 8 bit	1 MHz 8 bit	1.2 MHz 8 bit	3.6 MHz 16 bit
RAM	8 KB	2 KB	64 KB	0.125 KB	128 KB

	Game Boy	NES	Commodore 64	Atari 2600	SNES
CPU	1 MHz* 8 bit	1.8 MHz 8 bit	1 MHz 8 bit	1.2 MHz 8 bit	3.6 MHz 16 bit
RAM	8 KB	2 KB	64 KB	0.125 KB	128 KB
VRAM	8 KB	2 KB	16 KB (shared)	0	64 KB

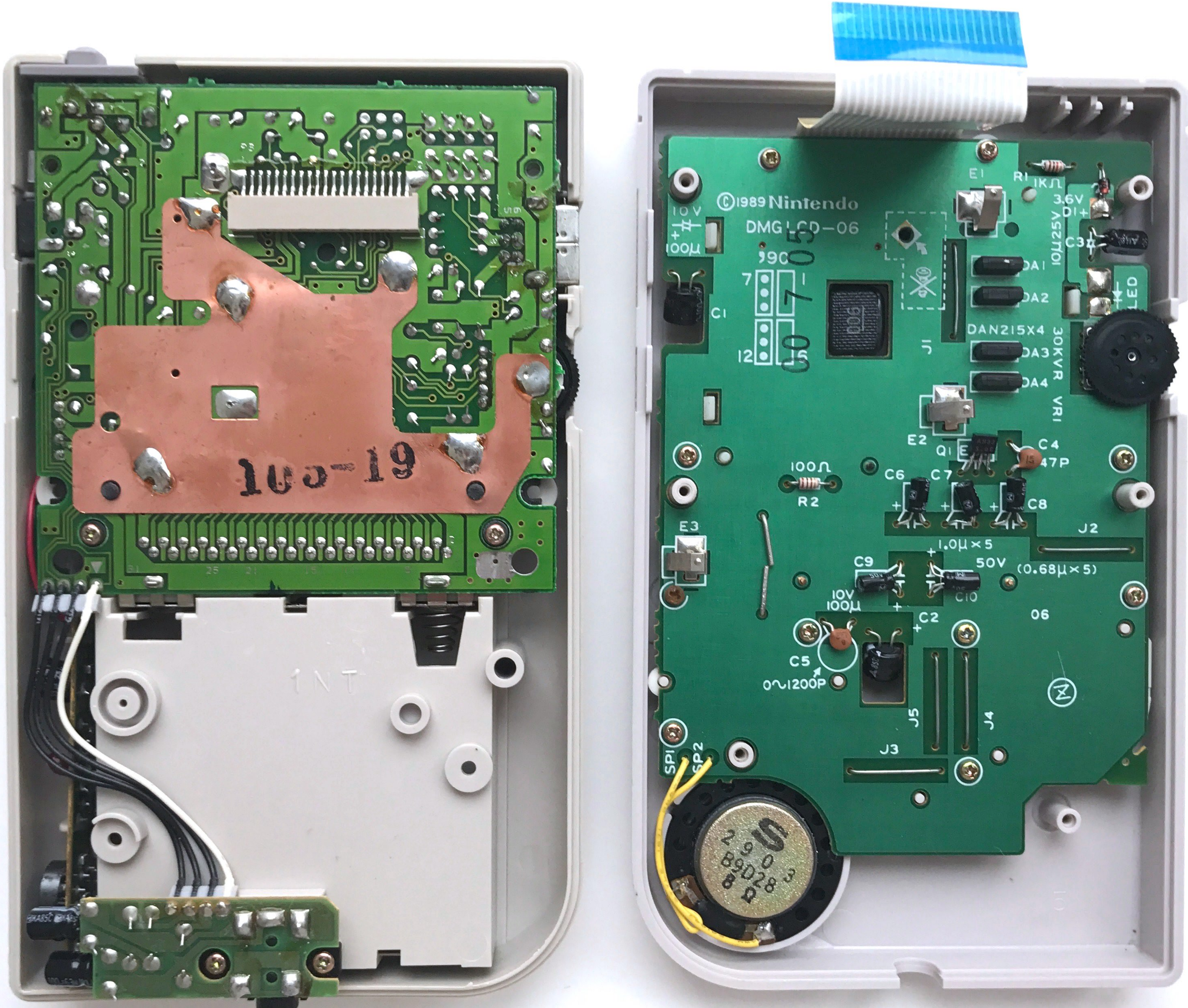
	Game Boy	NES	Commodore 64	Atari 2600	SNES
CPU	1 MHz* 8 bit	1.8 MHz 8 bit	1 MHz 8 bit	1.2 MHz 8 bit	3.6 MHz 16 bit
RAM	8 KB	2 KB	64 KB	0.125 KB	128 KB
VRAM	8 KB	2 KB	16 KB (shared)	0	64 KB
Resolution	160x144	256x240	320x200	160x192	256x224 512x448

	Game Boy	NES	Commodore 64	Atari 2600	SNES
CPU	1 MHz* 8 bit	1.8 MHz 8 bit	1 MHz 8 bit	1.2 MHz 8 bit	3.6 MHz 16 bit
RAM	8 KB	2 KB	64 KB	0.125 KB	128 KB
VRAM	8 KB	2 KB	16 KB (shared)	0	64 KB
Resolution	160x144	256x240	320x200	160x192	256x224 512x448
Simultaneous Colors	4	25	16	128	32768

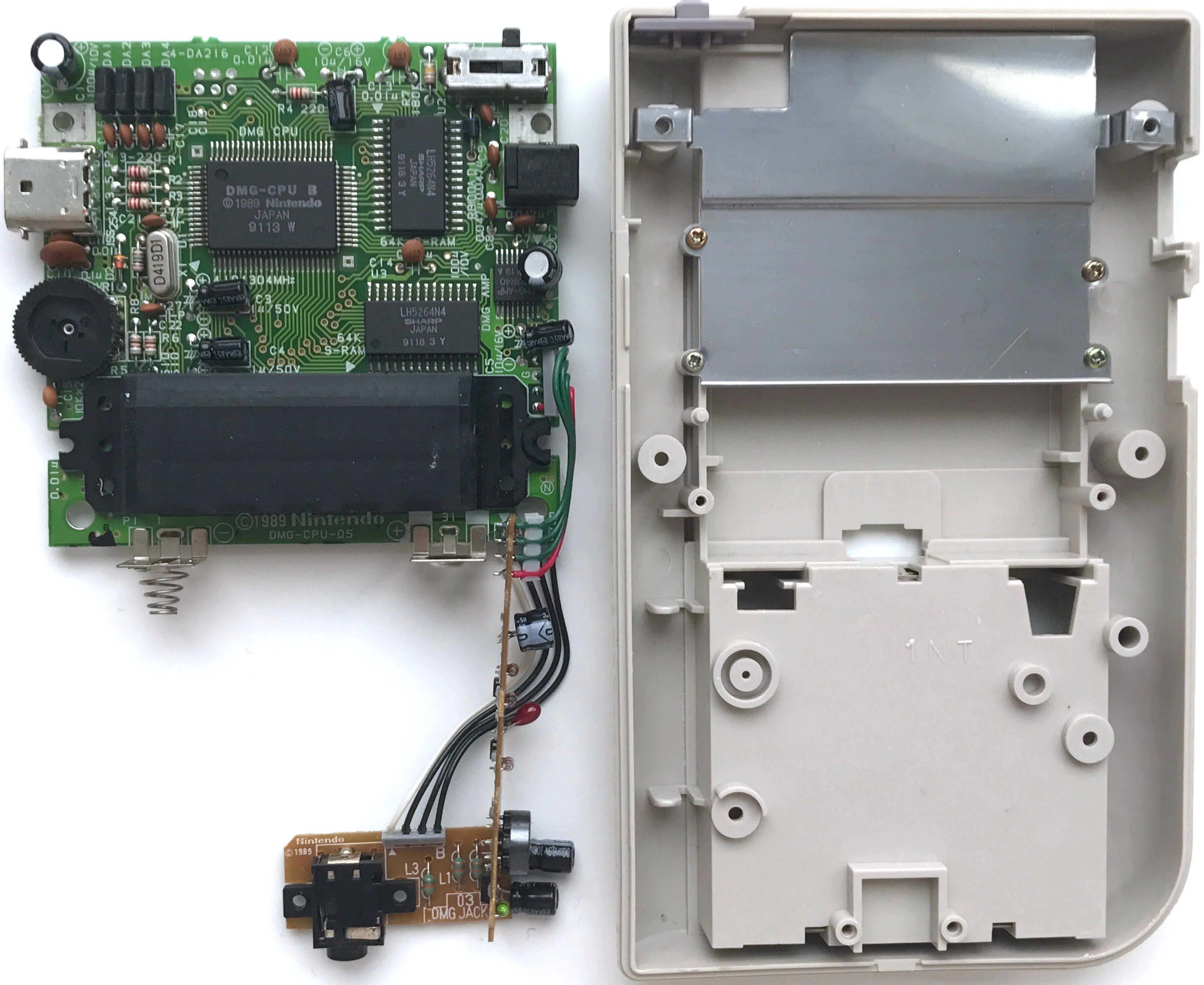
	Game Boy	NES	Commodore 64	Atari 2600	SNES
CPU	1 MHz* 8 bit	1.8 MHz 8 bit	1 MHz 8 bit	1.2 MHz 8 bit	3.6 MHz 16 bit
RAM	8 KB	2 KB	64 KB	0.125 KB	128 KB
VRAM	8 KB	2 KB	16 KB (shared)	0	64 KB
Resolution	160x144	256x240	320x200	160x192	256x224 512x448
Simultaneous Colors	4	25	16	128	32768
Sprites/Line	10	8	8	3	32

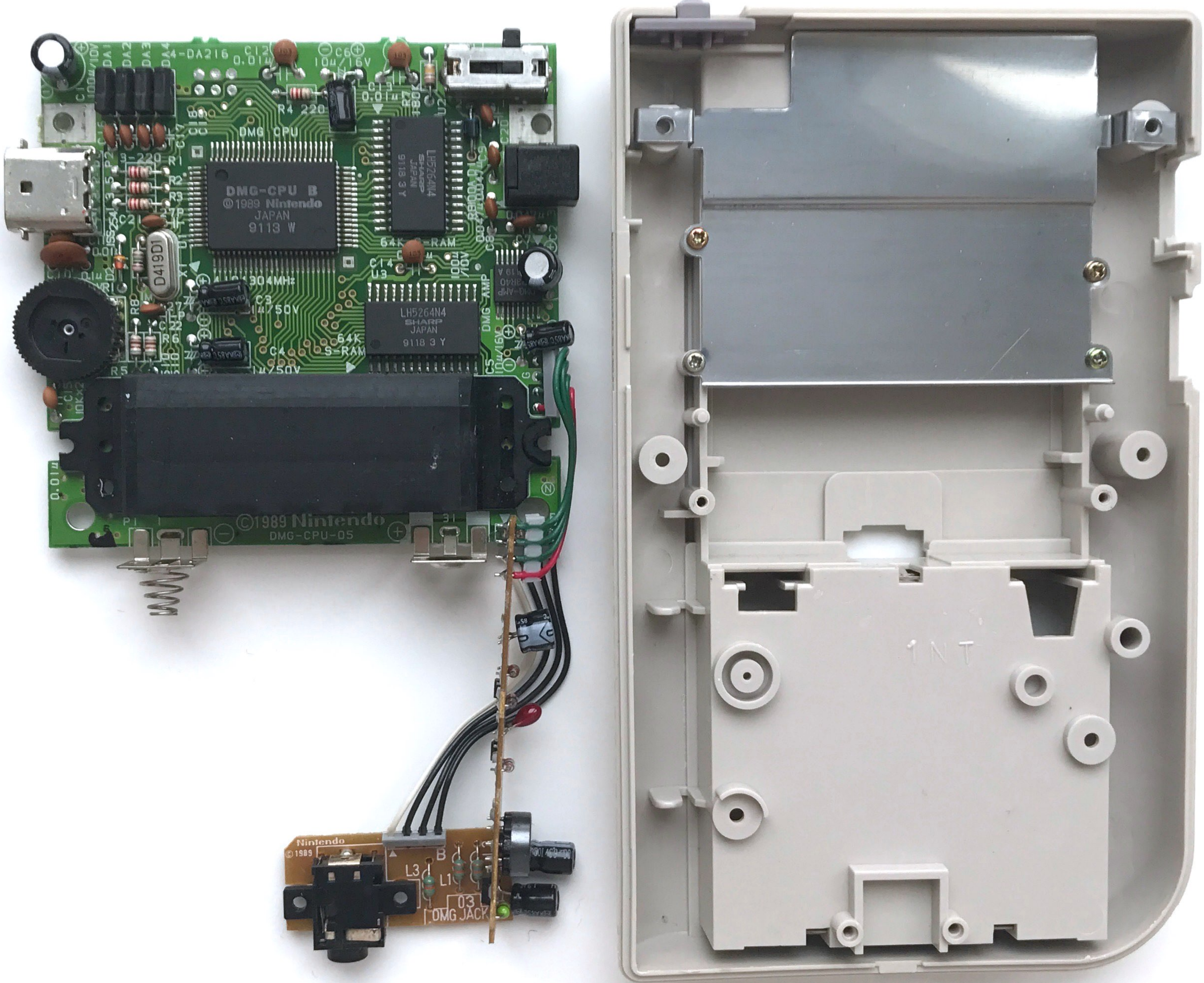
	Game Boy	NES	Commodore 64	Atari 2600	SNES
CPU	1 MHz* 8 bit	1.8 MHz 8 bit	1 MHz 8 bit	1.2 MHz 8 bit	3.6 MHz 16 bit
RAM	8 KB	2 KB	64 KB	0.125 KB	128 KB
VRAM	8 KB	2 KB	16 KB (shared)	0	64 KB
Resolution	160x144	256x240	320x200	160x192	256x224 512x448
Simultaneous Colors	4	25	16	128	32768
Sprites/Line	10	8	8	3	32
Year	**1989**	**1983**	**1982**	**1977**	**1990**


OFF•ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY™
B
A
SELECT
START
PHONES



©1989 Nintendo
DMG LCD-06
100-19
DAN215X4
1.0μ×5
50V
(0.68μ×5)
0~1200P


DMG CPU
DMG-CPU B
©1989 Nintendo
JAPAN
9113 W
LH5264N4
SHARP
JAPAN
9118 3 Y
64K S-RAM
©1989 Nintendo
DMG-CPU-05
DMG JACK


DMG CPU
DMG-CPU B
©1989 Nintendo
JAPAN
9113 W
LH5264N4
SHARP
JAPAN
9118 3 Y
64K S-RAM
DMG-AMP
©1989 Nintendo
DMG-CPU-05
Nintendo
©1989
DMG JACK


DMG-CPU B
©1989 Nintendo
JAPAN
9113 W
DMG CPU
LH5264N4
SHARP
JAPAN
9118 3 Y
64K S-RAM
LH5264N4
SHARP
JAPAN
9118 3 Y
64K
S-RAM
4.194304MHz
D419D1
C3
1μ/50V
C4
1μ/50V
R4 220
C12
0.01μ
C6
10μ/16V
C13
0.01μ
C14
U3
DMG AMP
DMG-AMP
C5
10μ/16V
R1
R2
R3
R5
R6
R8
C22
C21
C10
C11
VR1
4-DA216
3-220
100μ/10V

DMG CPU
DMG-CPU B
©1989 Nintendo
JAPAN
9113 W
LH5264N4
SHARP
JAPAN
9118 3 Y
64K S-RAM
LH5264N4
SHARP
JAPAN
9118 3 Y
64K
S-RAM
DMG AMP
4.194304MHz
D419D1
C3
1μ/50V
C4
1μ/50V
R4 220
C12
0.01μ
C6
10μ/16V
C13
0.01μ
4-DA216
3-220
R1
R2
R3
C14
U3
R5
R6
R8
C22
C5
10μ/16V

Game Boy
DMG


RAM
VRAM
DMG-CPU B
©1989 Nintendo
JAPAN
9113 W
LH5264N4
SHARP
JAPAN
9118 3 Y
DMG CPU
64K S-RAM
4.194304MHz
D419D1

Game Boy
DMG

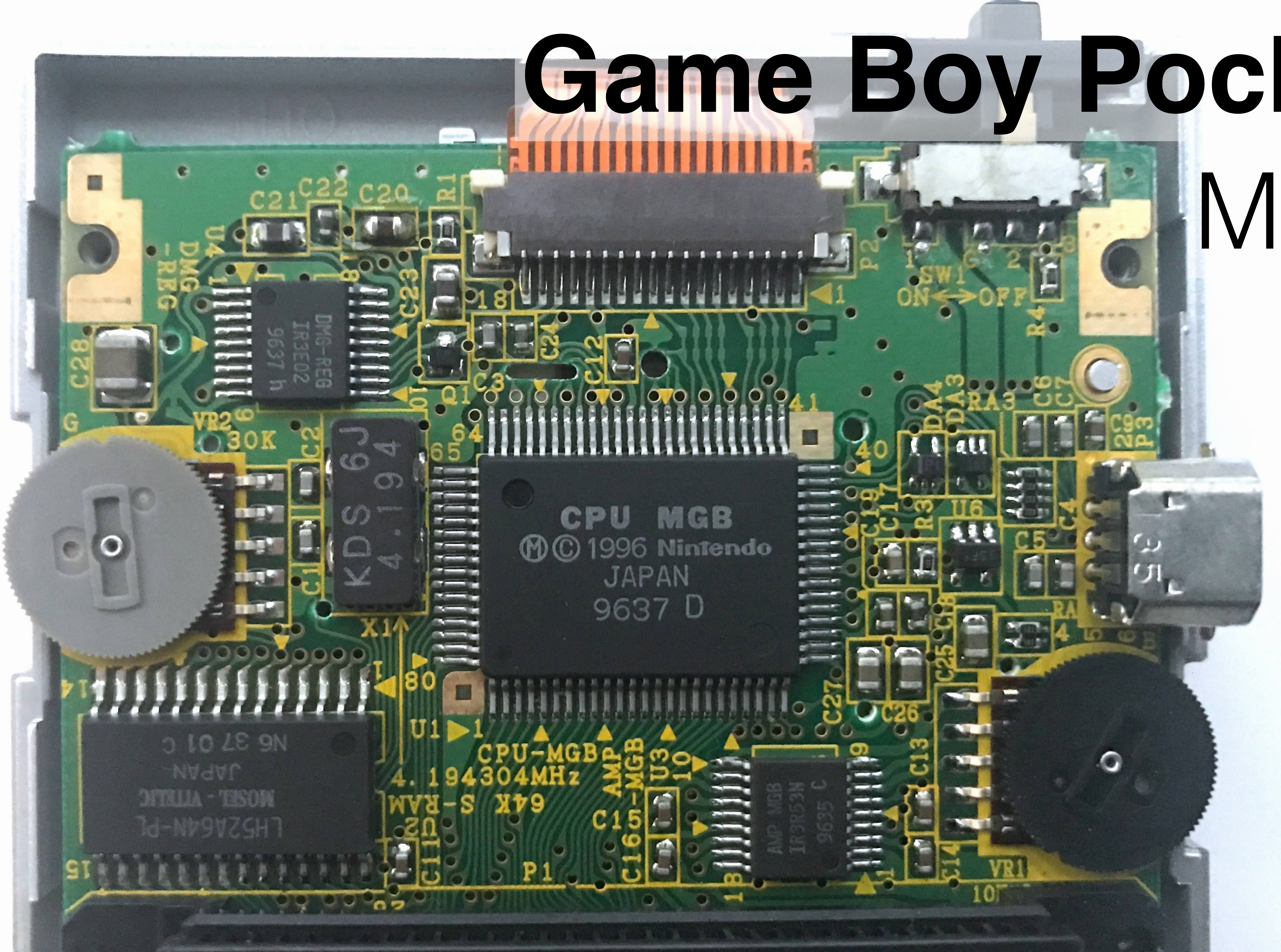

RAM
SoC
VRAM
DMG CPU
DMG-CPU B
©1989 Nintendo
JAPAN
9113 W
LH5264N4
SHARP
JAPAN
9118 3 Y
LH5264N4
SHARP
JAPAN
9118 3 Y
64K S-RAM
64K S-RAM
4.194304MHz
D419D1
DMG-AMP

Super Game Boy

SGB

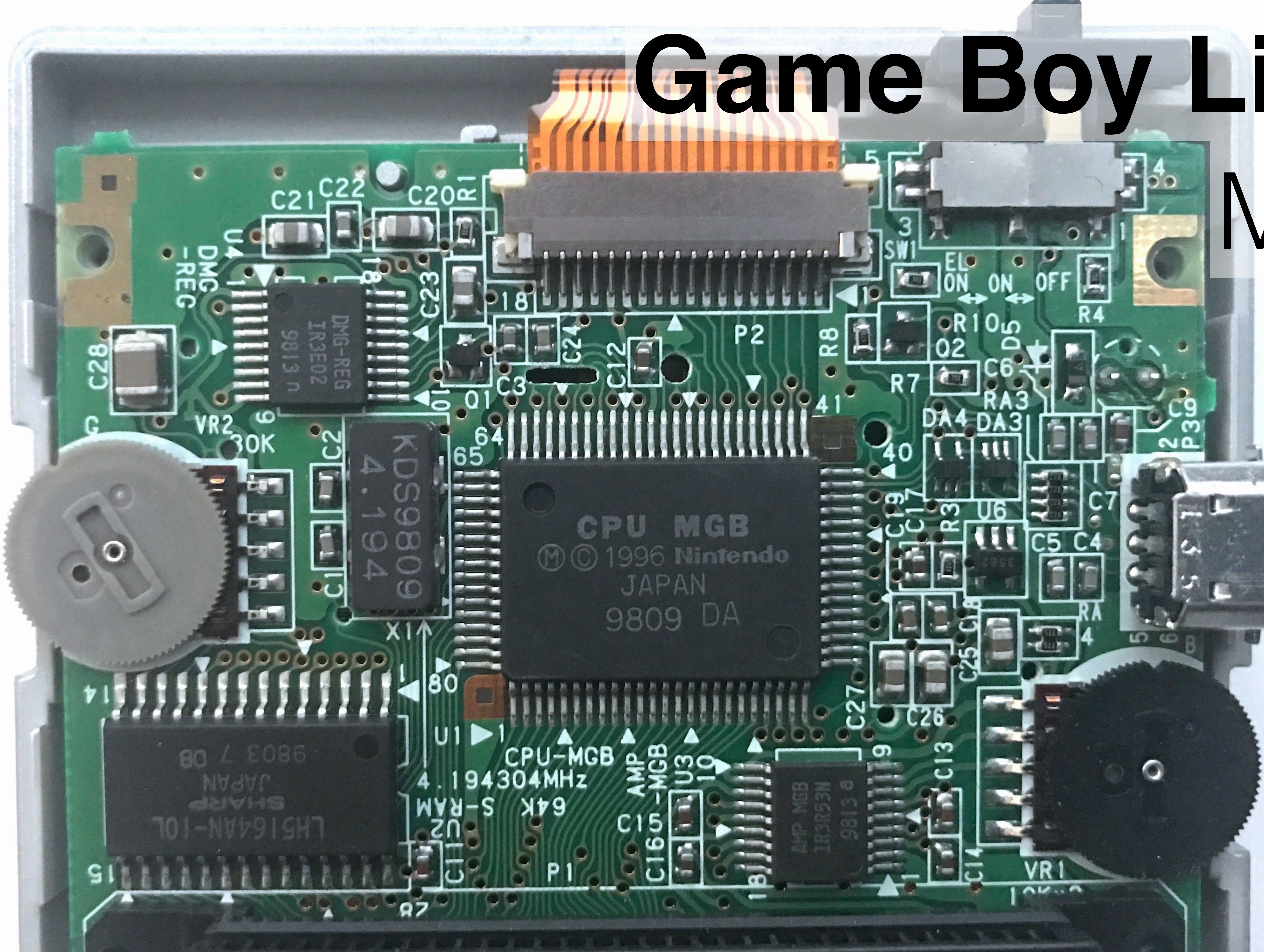
Game Boy Pocket

MGB



Game Boy Light

MGB

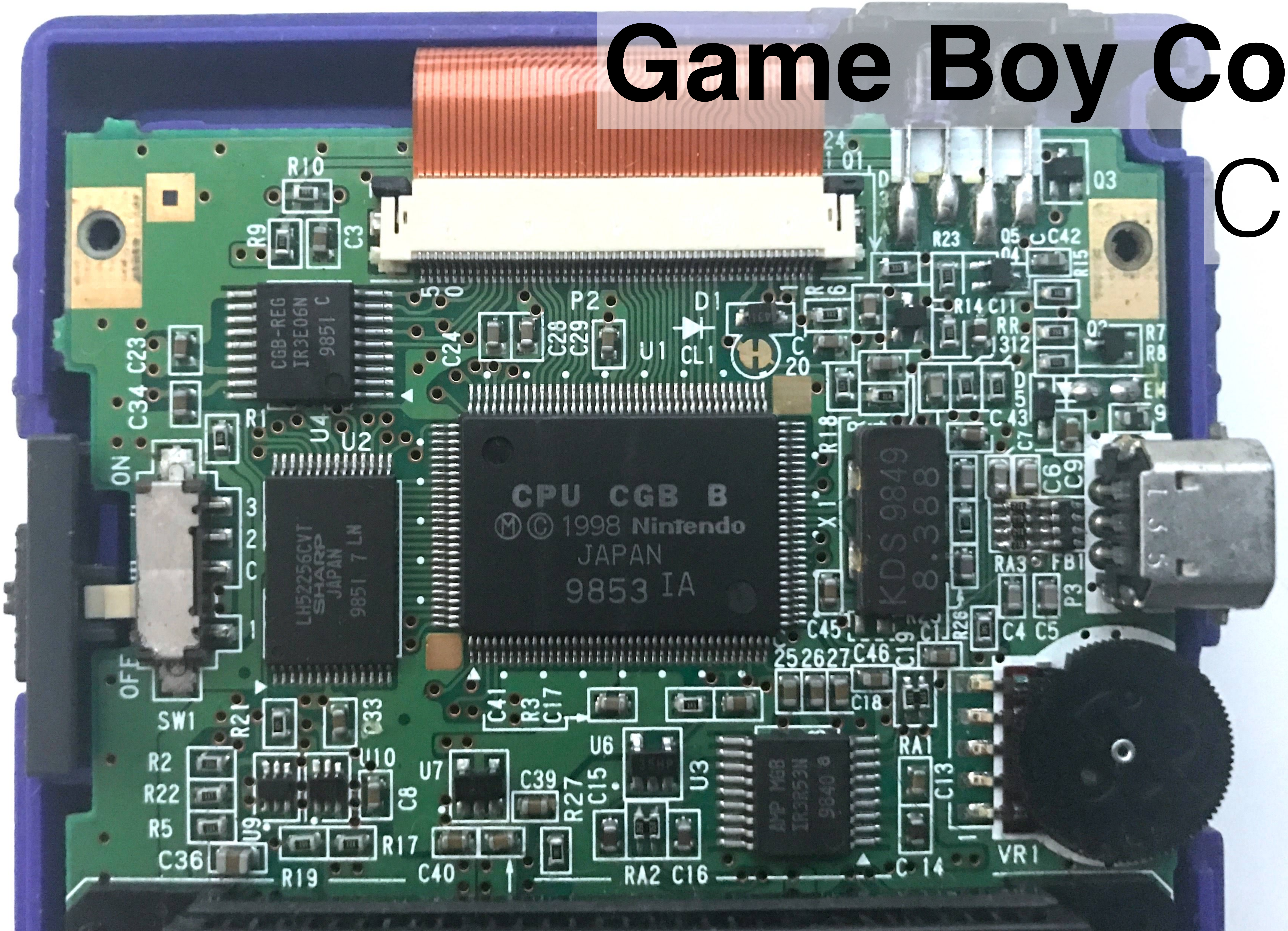


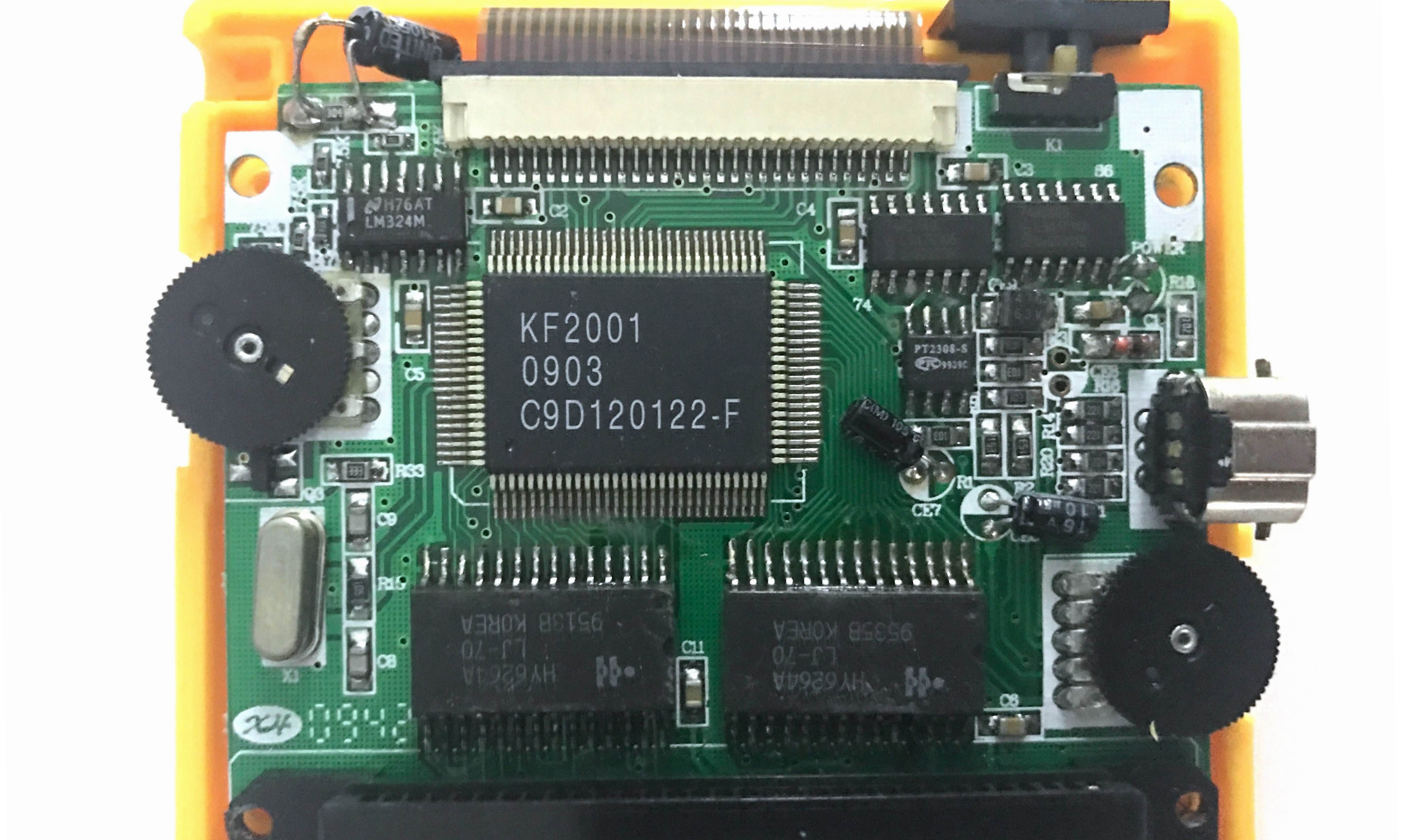
Super Game Boy 2

SGB2

Game Boy Color

CGB




KF2001
0903
C9D120122-F
H76AT
LM324M
PT2308-S
POWER
C2
C4
C5
C9
C8
C11
R33
CE7
R1
R2
K1


GB Boy
KF2001
0903
C9D120122-F
H76AT
LM324M
PT2308-S


OFF ON
POWER
GB BOY
B
A
SELECT START


RAM
SoC
VRAM
DMG-CPU B
©1989 Nintendo
JAPAN
9113 W
LH5264N4
SHARP
JAPAN
9118 3 Y
DMG CPU
64K S-RAM
4.194304MHz


DMG-CPU B
©1989 Nintendo
JAPAN
9113 W

DMG–CPU

CPU

Interrupt Controller

Timer

Memory

Boot ROM

Joypad Input

Serial Data Transfer

Sound Controller

Pixel Processing Unit

CPU

Game Boy

1989

Wikimedia Commons

NES
1983

Game Boy
1989

Wikimedia Commons

NES
1983

Game Boy
1989

SNES
1990

Wikimedia Commons

NES
1983

Game Boy
1989

SNES
1990

6502

Wikimedia Commons

NES
1983

6502

Game Boy
1989

SNES
1990

65816

Wikimedia Commons

NES
1983

MOS
6502

Game Boy
1989

SHARP
LR35902

SNES
1990

WDC
65816

Wikimedia Commons

1989

SHARP
LR35902

Wikimedia Commons
http://www.oldcomputers.net/

1974	1989	1976
intel 8080	SHARP LR35902	Zilog Z80

Wikimedia Commons
http://www.oldcomputers.net/

1974

intel

8080

Altair 8800

IMSAI 8080

1989

SHARP

LR35902

1976

Zilog

Z80

Wikimedia Commons

http://www.oldcomputers.net/

1974	1989	1976
intel 8080	SHARP LR35902	Zilog Z80

Altair 8800

IMSAI 8080

Osborne 1

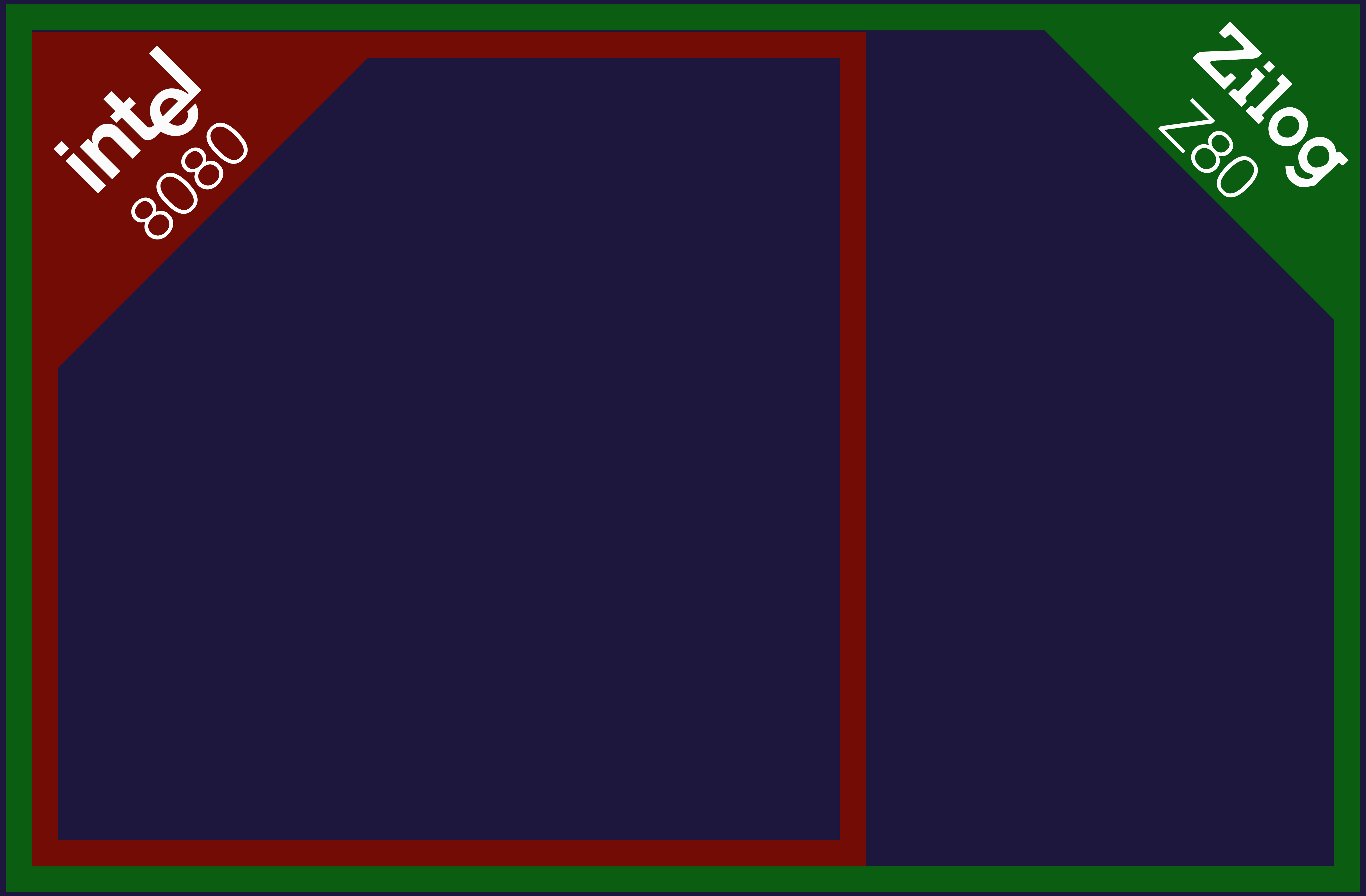
TRS-80

Spectrum

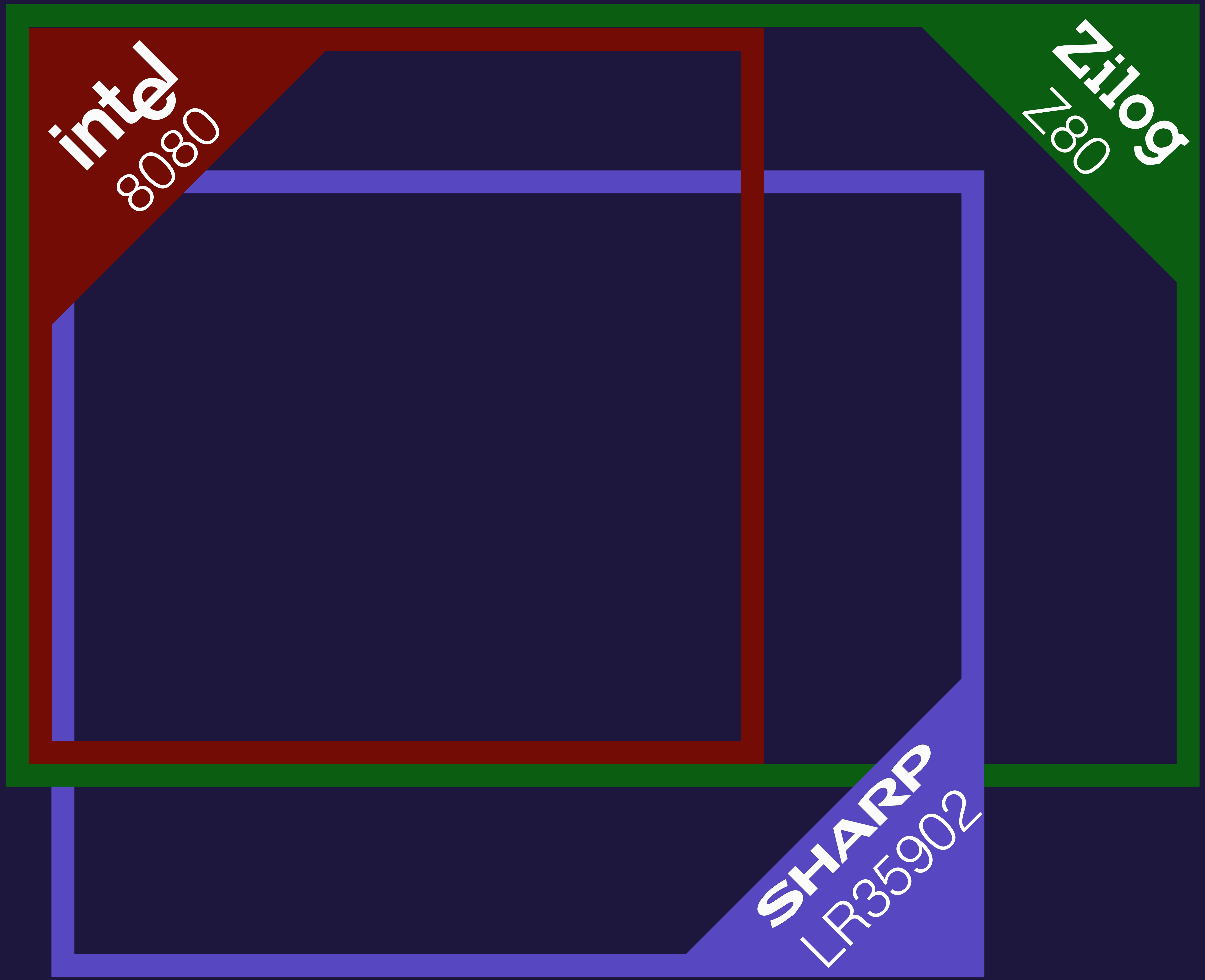
Wikimedia Commons

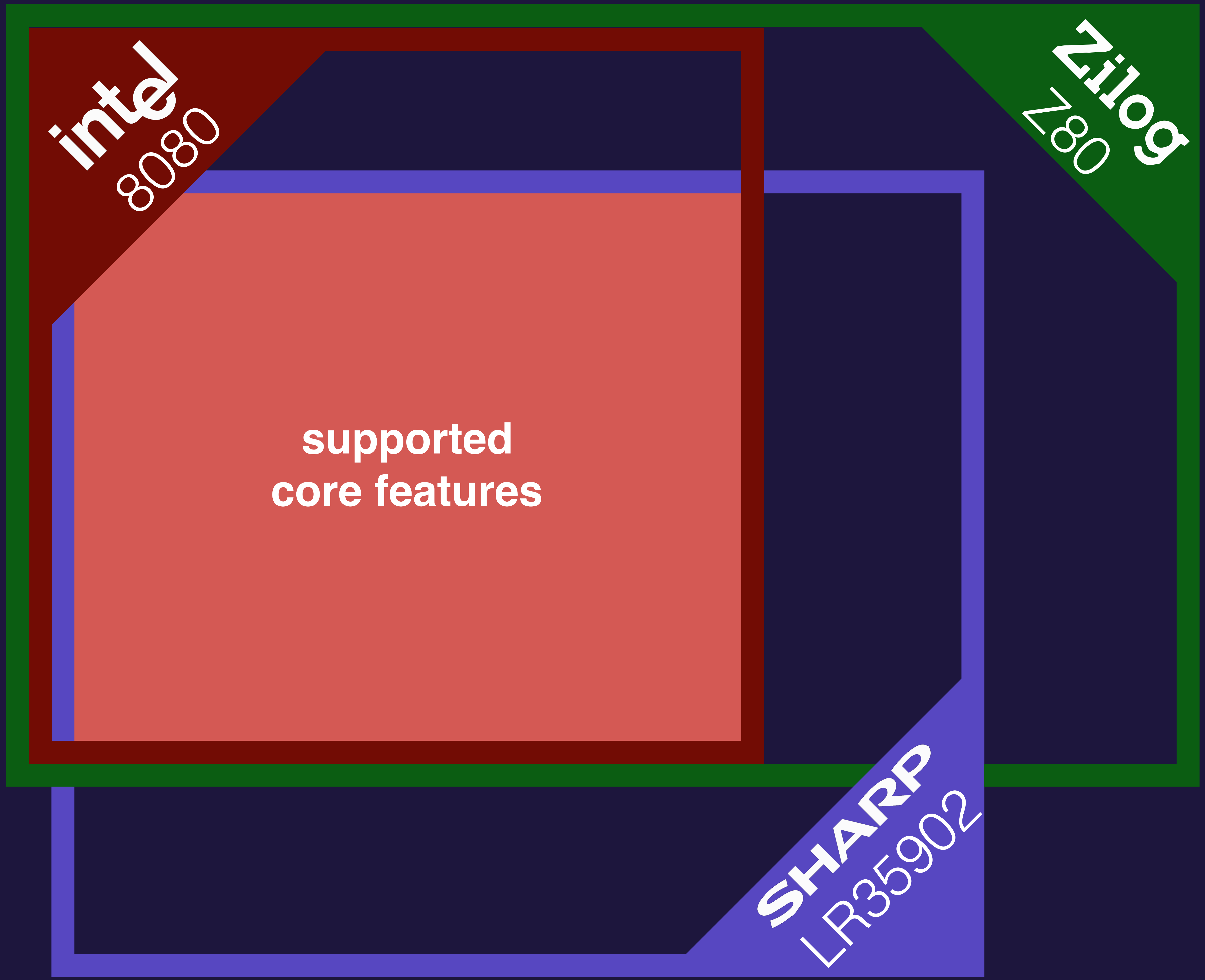
http://www.oldcomputers.net/

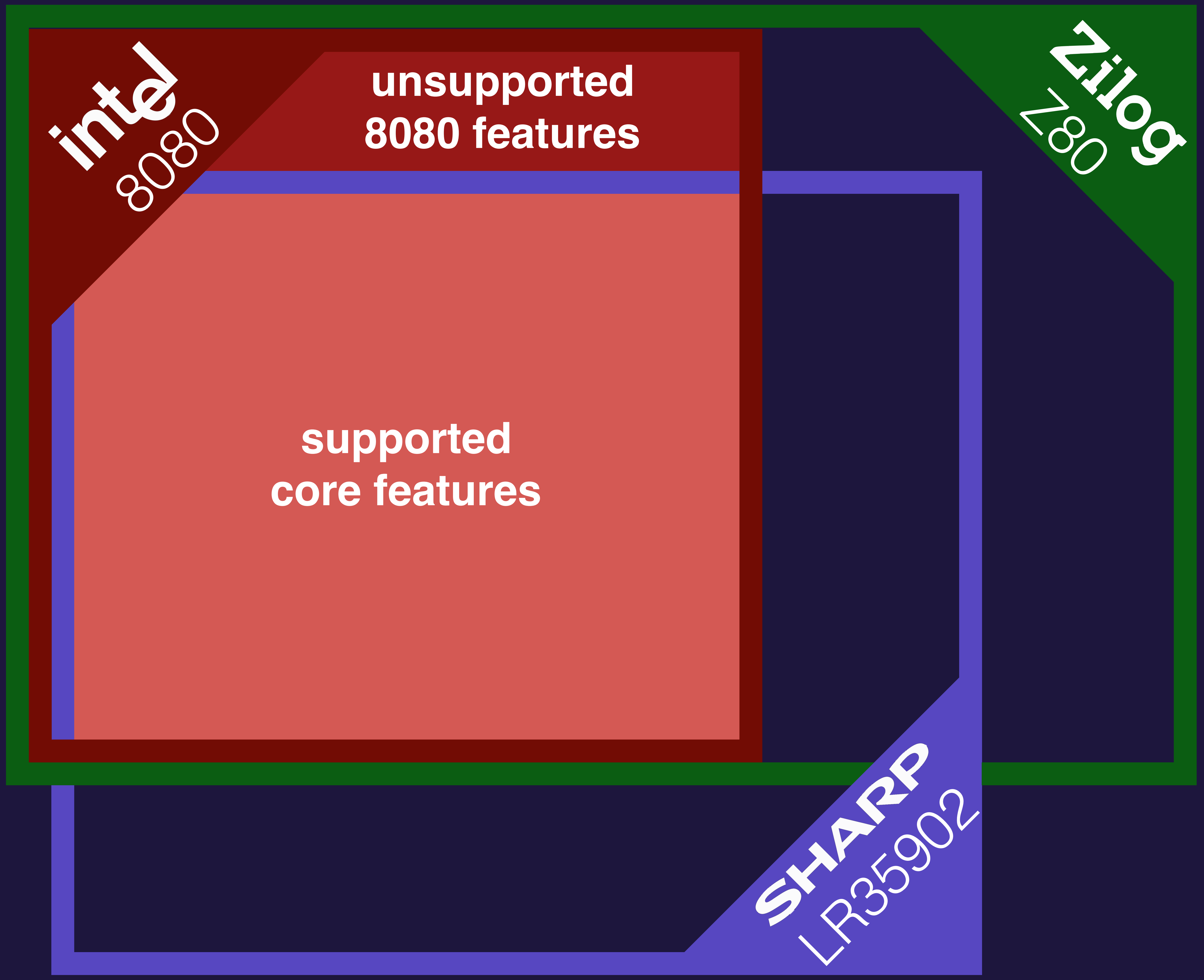
intel
8080

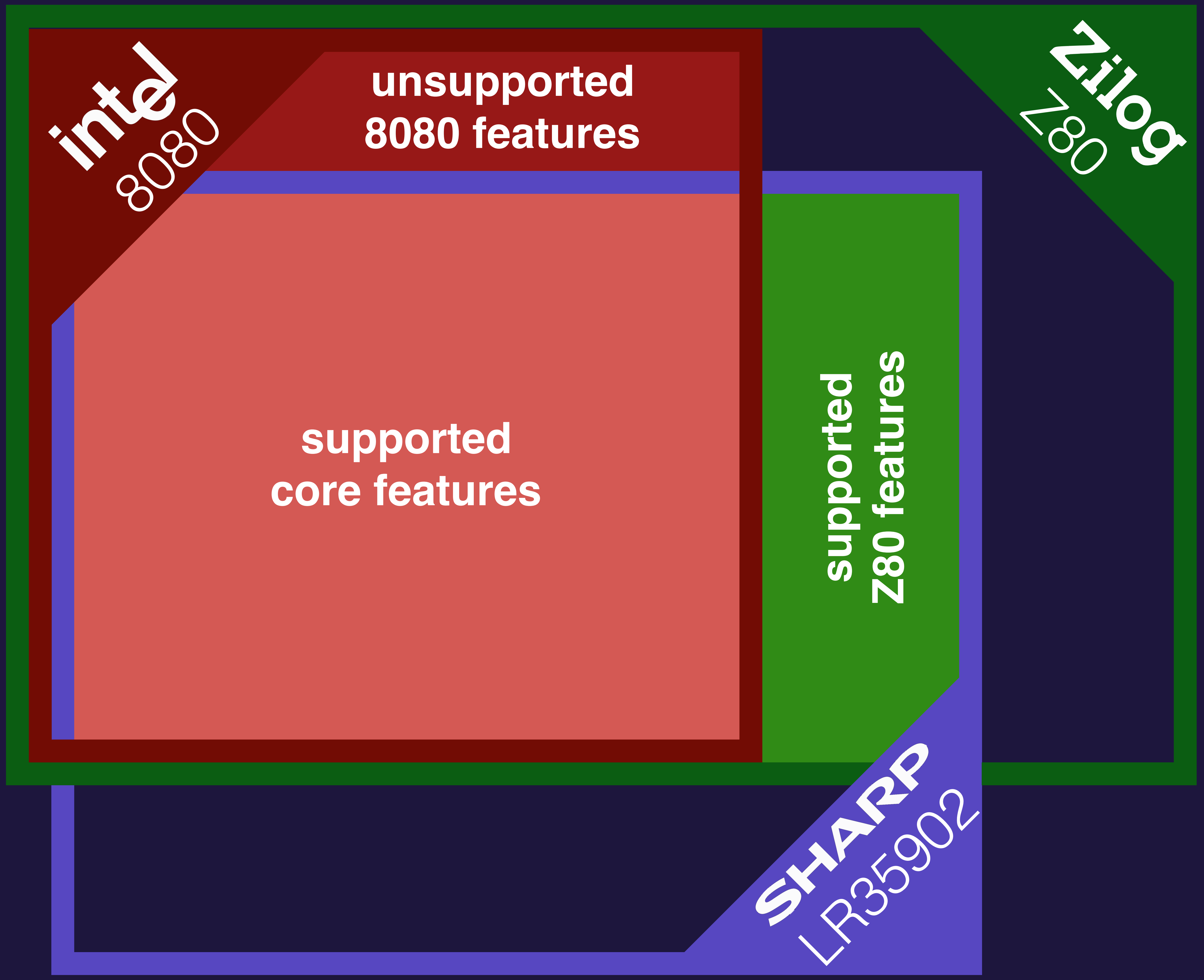


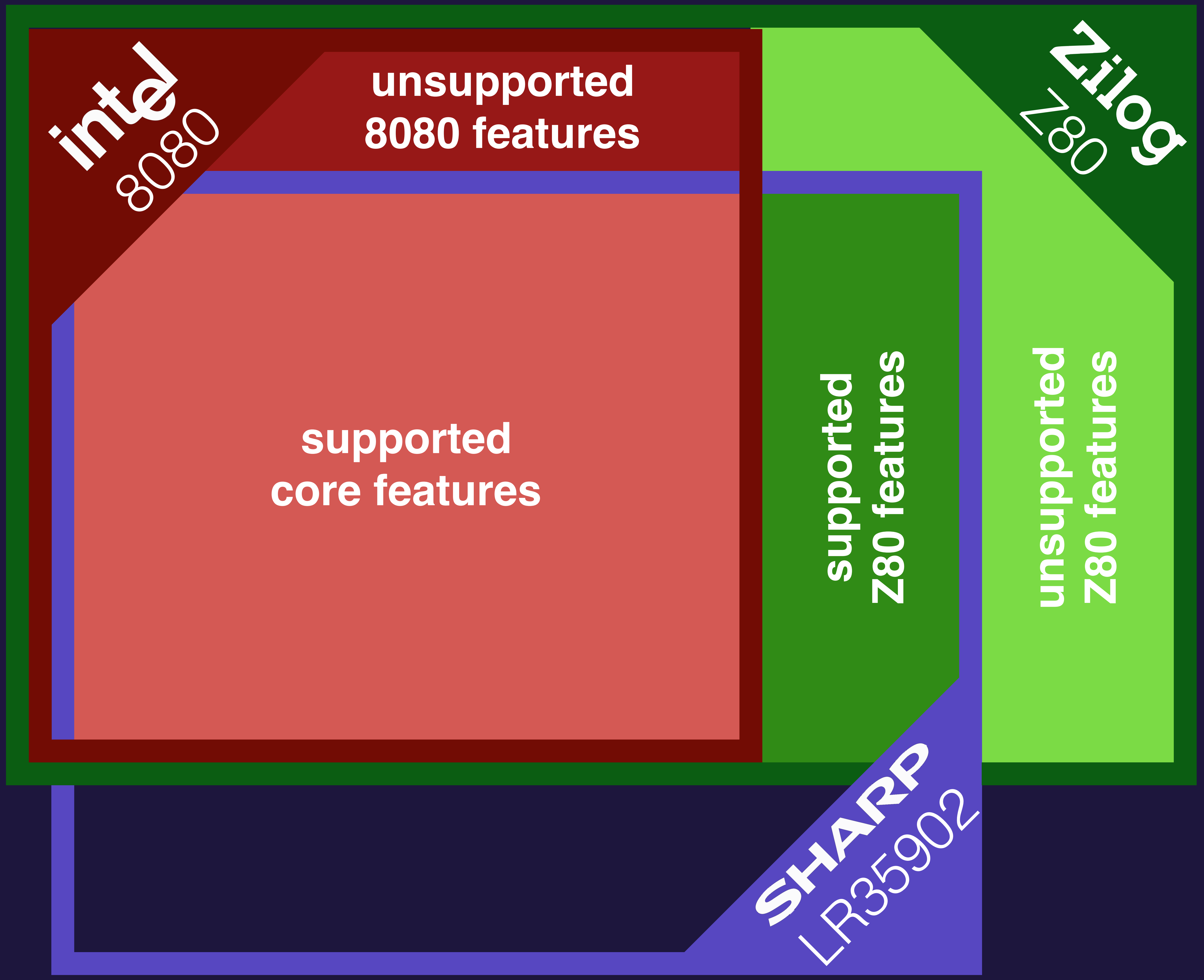
intel
8080
Zilog
Z80

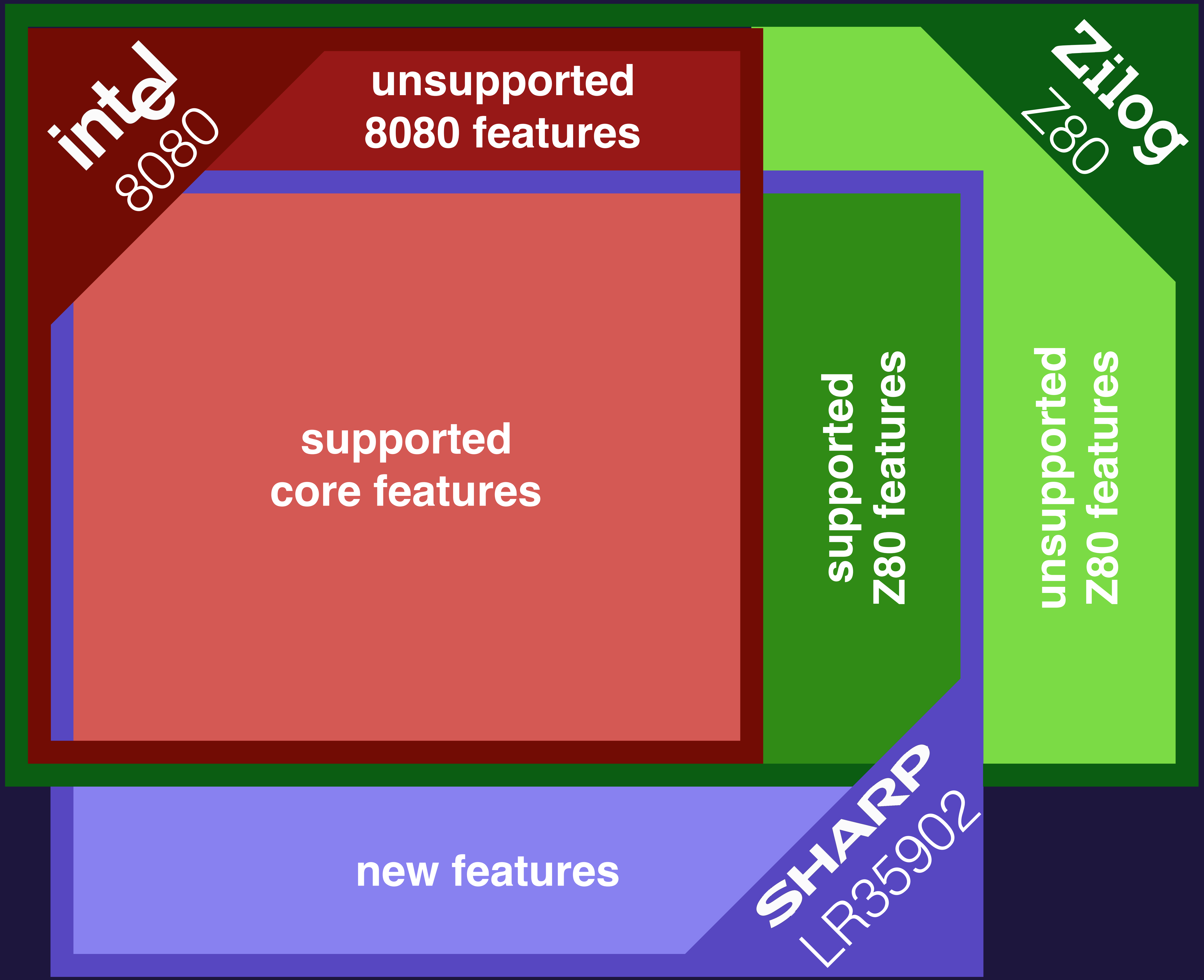

intel
8080
Zilog
Z80
SHARP
LR35902


intel
8080
Zilog
Z80
SHARP
LR35902
supported
core features


unsupported
8080 features
supported
core features
8080
Z80
LR35902


intel
8080
unsupported
8080 features
Zilog
Z80
supported
core features
supported
Z80 features
SHARP
LR35902


intel
8080
unsupported
8080 features
supported
core features
supported
Z80 features
unsupported
Z80 features
Zilog
Z80
SHARP
LR35902


intel
8080
unsupported
8080 features
Zilog
Z80
supported
core features
supported
Z80 features
unsupported
Z80 features
new features
SHARP
LR35902

Supported Core Features

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

Flags *							
Z	N	H	C	-	-	-	-

* different layout than 8080

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

Flags *

Z	N	H	C	-	-	-	-

* different layout than 8080

Supported Core Features

Registers	
A	F
BC	
D	E
H	L
SP	
PC	

Flags *							
Z	N	H	C	-	-	-	-

* different layout than 8080

Supported Core Features

Registers	
A	F
B	C
DE	
H	L
SP	
PC	

Flags *							
Z	N	H	C	-	-	-	-

* different layout than 8080

Supported Core Features

Registers	
A	F
B	C
D	E
HL	
SP	
PC	

Flags *							
Z	N	H	C	-	-	-	-

* different layout than 8080

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

r16
bc
de
hl
sp

Flags*							
Z	N	H	C	-	-	-	-

* different layout than 8080

Supported Core Features

r8
a
b
c
d
e
h
l
(hl)

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

r16
bc
de
hl
sp

Flags *							
Z	N	H	C	-	-	-	-

* different layout than 8080

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

r8
a (hl)
b c
d e
h l

r16
bc
de
hl
sp

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

Load/Store

```
ld r8, d8
ld r8, r8
ld a, (bc)
ld a, (de)
ld a, (a16)*
ld (bc), a
ld (de), a
ld (a16), a*
ld r16, d16
```

r8

```
a (hl)
b c
d e
h l
```

r16

```
bc
de
hl
sp
```

* different opcode than 8080

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

r8	
a	(hl)
b	c
d	e
h	l

r16
bc
de
hl
sp

Load/Store

```
ld r8, d8
ld r8, r8
ld a, (bc)
ld a, (de)
ld a, (a16)*
ld (bc), a
ld (de), a
ld (a16), a*
ld r16, d16
```

Stack

```
ld sp, hl
push r16
pop r16
```

* different opcode than 8080

Supported Core Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

r8
a (hl)
b c
d e
h l

r16
bc
de
hl
sp

Load/Store

```
ld r8, d8
ld r8, r8
ld a, (bc)
ld a, (de)
ld a, (a16)*
ld (bc), a
ld (de), a
ld (a16), a*
ld r16, d16
```

Stack

```
ld sp, hl
push r16
pop r16
```

Artih/Logical

```
add |
adc |
sub |  a, d8
sbc |  a, r8
and |
xor |
or  |
cp  |
inc |  r8
dec |
daa
cpl
add hl, r16
inc r16
dec r16
```

* different opcode than 8080

Supported Core Features

Registers

A	F
B	C
D	E
H	L
SP	
PC	

r8

```
a (hl)
b c
d e
h l
```

r16

```
bc
de
hl
sp
```

Load/Store

```
ld r8, d8
ld r8, r8
ld a, (bc)
ld a, (de)
ld a, (a16)*
ld (bc), a
ld (de), a
ld (a16), a*
ld r16, d16
```

Stack

```
ld sp, hl
push r16
pop r16
```

Artih/Logical

```
add |
adc |
sub | a, d8
sbc | a, r8
and |
xor |
or  |
cp  |
inc | r8
dec |
daa
cpl
add hl, r16
inc r16
dec r16
```

Rotate

```
rlca
rla
rrca
rra
```

* different opcode than 8080

Supported Core Features

Registers

A	F
B	C
D	E
H	L
SP	
PC	

r8

```
a (hl)
b c
d e
h l
```

r16

```
bc
de
hl
sp
```

Load/Store

```
ld r8, d8
ld r8, r8
ld a, (bc)
ld a, (de)
ld a, (a16)*
ld (bc), a
ld (de), a
ld (a16), a*
ld r16, d16
```

Stack

```
ld sp, hl
push r16
pop r16
```

Artih/Logical

```
add | a, d8
adc | a, r8
sub |
sbc |
and |
xor |
or  |
cp  |
inc | r8
dec |
daa
cpl
add hl, r16
inc r16
dec r16
```

Rotate

```
rlca
rla
rrca
rra
```

Control Flow

```
jp a16
jp hl
jp f, a16
call a16
call f, a16
ret
ret f
rst n
```

* different opcode than 8080

Supported Core Features

Registers

A	F
B	C
D	E
H	L
SP	
PC	

r8

a (hl)
b c
d e
h l

r16

bc
de
hl
sp

Load/Store

```
ld r8, d8
ld r8, r8
ld a, (bc)
ld a, (de)
ld a, (a16)*
ld (bc), a
ld (de), a
ld (a16), a*
ld r16, d16
```

Stack

```
ld sp, hl
push r16
pop r16
```

Artih/Logical

add / adc / sub / sbc / and / xor / or / cp | a, d8 / a, r8

inc / dec | r8

```
daa
cpl
add hl, r16
inc r16
dec r16
```

Rotate

```
rlca
rla
rrca
rra
```

Control Flow

```
jp a16
jp hl
jp f, a16
call a16
call f, a16
ret
ret f
rst n
```

Misc

```
ccf
scf
nop
halt
di
ei
```

* different opcode than 8080

Supported Core Features

Supported Core Features

0000
0008
0010
0018
0020
0028
0030
0038
0040
0048
0050
0058
0060
.
.
.

Supported Core Features

```
0000
0008
0010
0018
0020
0028
0030
0038
0040    irq service routine #0
0048    irq service routine #1
0050    irq service routine #2
0058    irq service routine #3
0060    irq service routine #4
  .
  .
  .
```

Supported Core Features

```
0000    rst $00
0008    rst $08
0010    rst $10
0018    rst $18
0020    rst $20
0028    rst $28
0030    rst $30
0038    rst $38
0040    irq service routine #0
0048    irq service routine #1
0050    irq service routine #2
0058    irq service routine #3
0060    irq service routine #4
  .
  .
  .
```

Supported Core Features

```
0000    rst $00  <------  RESET
0008    rst $08
0010    rst $10
0018    rst $18
0020    rst $20
0028    rst $28
0030    rst $30
0038    rst $38
0040    irq service routine #0
0048    irq service routine #1
0050    irq service routine #2
0058    irq service routine #3
0060    irq service routine #4
  .
  .
  .
```

Unsupported 8080 Features

Unsupported 8080 Features

Flags							
Z	N	H	C	-	-	-	-

Unsupported 8080 Features

Flags							
S	Z	-	H	-	P	-	C

Unsupported 8080 Features

Flags							
S	Z	-	H	-	P	-	C

Unsupported 8080 Features

Flags							
S	Z	-	H	-	P	-	C

Sign & Parity

```
ret po
jp   |
call | po, nn
ret pe
jp   |
call | pe, nn
ret p
jp   |
call | p, nn
ret m
jp   |
call | m, nn
```

Unsupported 8080 Features

Flags							
S	Z	-	H	-	P	-	C

Sign & Parity

```
ret po
jp   | po, nn
call |
ret pe
jp   | pe, nn
call |
ret p
jp   | p, nn
call |
ret m
jp   | m, nn
call |
```

Load/Store HL

```
ld (a16), hl
ld hl, (a16)
```

Unsupported 8080 Features

Flags

S	Z	-	H	-	P	-	C

Sign & Parity

```
ret po
jp   | po, nn
call |
ret pe
jp   | pe, nn
call |
ret p
jp   | p, nn
call |
ret m
jp   | m, nn
call |
```

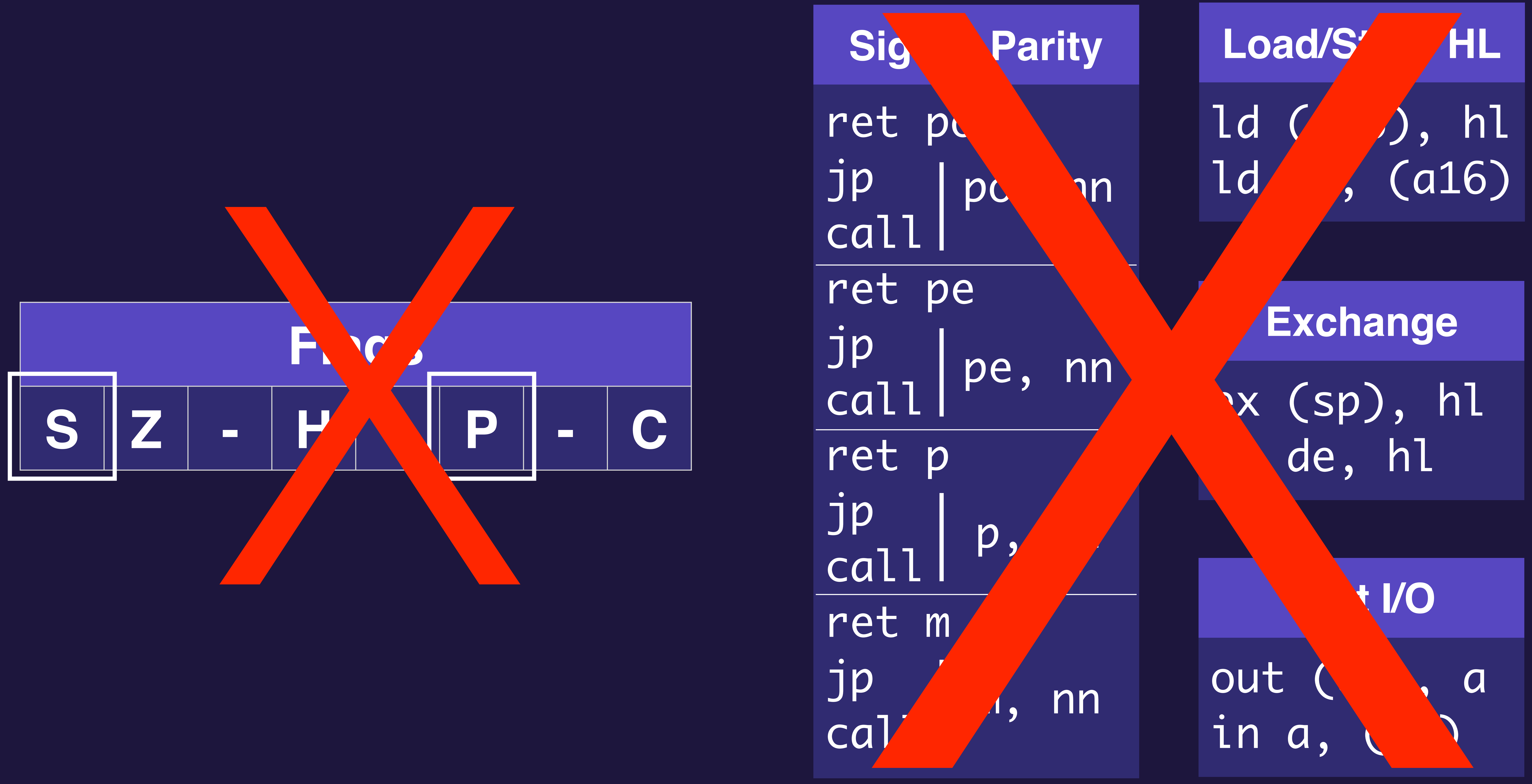
Load/Store HL

```
ld (a16), hl
ld hl, (a16)
```

Exchange

```
ex (sp), hl
ex de, hl
```

Unsupported 8080 Features



Supported Z80 Features

Supported Z80 Features

Shift & Bit Ops

```
rlc r8
rl r8
rrc r8
rr r8
sla r8
sra r8
srl r8
bit n, r8
set n, r8
res n, r8
```

CB Prefix

Supported Z80 Features

Shift & Bit Ops

```
rlc r8
rl  r8
rrc r8
rr  r8
sla r8
sra r8
srl r8
bit n, r8
set n, r8
res n, r8
```

CB Prefix

Relative Jump

```
jr a8
jr nz, a8
jr z, a8
jr nc, a8
jr c, a8
```

Supported Z80 Features

Shift & Bit Ops

```
rlc r8
rl r8
rrc r8
rr r8
sla r8
sra r8
srl r8
bit n, r8
set n, r8
res n, r8
```

CB Prefix

Relative Jump

```
jr a8
jr nz, a8
jr z, a8
jr nc, a8
jr c, a8
```

Interrupt Return

```
reti*
```

* different opcode than Z80

Unsupported Z80 Features

Unsupported Z80 Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

Unsupported Z80 Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

A'	F'
B'	C'
D'	E'
H'	L'

Unsupported Z80 Features

Registers	
A	F
B	C
D	E
H	L
SP	
PC	

A’	F’
B’	C’
D’	E’
H’	L’

IX
IY

Unsupported Z80 Features

Auto-Inc/Dec, Loop

```
cpi/cpir/cpd/cpdr
 i/inir/ind/ind
o  /outir/outd  tr
ldi  dir/ldd/l  r
djnz
```

Co o ow

SPR

```
 m 0/1/2
ld a, i
ld i, a
ld a, r
ld r, a
```

Registers

A	F
B	C
D	E
H	L
SP	
PC	

IX	
IY	

A	F
B’	C’
D’	E’
H	L

Alternate Regs

```
ex af, af'
exx
```

ad/Store 1

```
ld  16, (  6)
ld (  6  r16
```

4 R te

```
rld
rrd
```

Arithmetic

```
adc hl, r16
sbc hl, r16
neg
```

New Features

New Features

Postinc/Predec

```
ld (hl+), a
ld a, (hl+)
ld (hl-), a
ld a, (hl-)
```

New Features

Postinc/Predec

ld (hl+), a	mem[hl] = a; hl++;
ld a, (hl+)	a = mem[hl]; hl++;
ld (hl-), a	hl--; mem[hl] = a;
ld a, (hl-)	hl--; a = mem[hl];

New Features

New Features

Zero Page

```
ld ($ff00+a8), a
ld a, ($ff00+a8)
ld ($ff00+c), a
ld a, ($ff00+c)
```

New Features

Zero Page

```
ld ($ff00+a8), a
ld a, ($ff00+a8)
ld ($ff00+c), a
ld a, ($ff00+c)
```

0 KB

64 KB

New Features

Zero Page

```
ld ($ff00+a8), a
ld a, ($ff00+a8)
ld ($ff00+c), a
ld a, ($ff00+c)
```

New Features

Zero Page

```
ld ($ff00+a8), a
ld a, ($ff00+a8)
ld ($ff00+c), a
ld a, ($ff00+c)
```

```
fa 40 ff     ld a, ($ff40)     4 cycles
```

New Features

Zero Page

```
ld ($ff00+a8), a
ld a, ($ff00+a8)
ld ($ff00+c), a
ld a, ($ff00+c)
```

```
fa 40 ff     ld a, ($ff40)     4 cycles
f0 40        ld a, ($ff40)     3 cycles
```

New Features

Stack

```
add sp, r8
ld hl, sp+r8
```

New Features

Stack

```
add sp, r8
ld hl, sp+r8
```

Store SP

```
ld (a16), sp*
```

* Z80 has a more generic version

New Features

Stack

```
add sp, r8
ld hl, sp+r8
```

Store SP

```
ld (a16), sp*
```

Swap Nibbles

```
swap r8
```

* Z80 has a more generic version

New Features

Stack

```
add sp, r8
ld hl, sp+r8
```

Store SP

```
ld (a16), sp*
```

Swap Nibbles

```
swap r8
```

Power Saving

```
stop
```

* Z80 has a more generic version

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (BC),A 1 8 - - - -	INC BC 1 8 - - - -	INC B 1 4 Z 0 H -	DEC B 1 4 Z 1 H -	LD B,d8 2 8 - - - -	RLCA 1 4 0 0 0 C	LD (a16),SP 3 20 - - - -	ADD HL,BC 1 8 - 0 H C	LD A,(BC) 1 8 - - - -	DEC BC 1 8 - - - -	INC C 1 4 Z 0 H -	DEC C 1 4 Z 1 H -	LD C,d8 2 8 - - - -	RRCA 1 4 0 0 0 C
1x	STOP 0 2 4 - - - -	LD DE,d16 3 12 - - - -	LD (DE),A 1 8 - - - -	INC DE 1 8 - - - -	INC D 1 4 Z 0 H -	DEC D 1 4 Z 1 H -	LD D,d8 2 8 - - - -	RLA 1 4 0 0 0 C	JR r8 2 12 - - - -	ADD HL,DE 1 8 - 0 H C	LD A,(DE) 1 8 - - - -	DEC DE 1 8 - - - -	INC E 1 4 Z 0 H -	DEC E 1 4 Z 1 H -	LD E,d8 2 8 - - - -	RRA 1 4 0 0 0 C
2x	JR NZ,r8 2 12/8 - - - -	LD HL,d16 3 12 - - - -	LD (HL+),A 1 8 - - - -	INC HL 1 8 - - - -	INC H 1 4 Z 0 H -	DEC H 1 4 Z 1 H -	LD H,d8 2 8 - - - -	DAA 1 4 Z - 0 C	JR Z,r8 2 12/8 - - - -	ADD HL,HL 1 8 - 0 H C	LD A,(HL+) 1 8 - - - -	DEC HL 1 8 - - - -	INC L 1 4 Z 0 H -	DEC L 1 4 Z 1 H -	LD L,d8 2 8 - - - -	CPL 1 4 - 1 1 -
3x	JR NC,r8 2 12/8 - - - -	LD SP,d16 3 12 - - - -	LD (HL-),A 1 8 - - - -	INC SP 1 8 - - - -	INC (HL) 1 12 Z 0 H -	DEC (HL) 1 12 Z 1 H -	LD (HL),d8 2 12 - - - -	SCF 1 4 - 0 0 1	JR C,r8 2 12/8 - - - -	ADD HL,SP 1 8 - 0 H C	LD A,(HL-) 1 8 - - - -	DEC SP 1 8 - - - -	INC A 1 4 Z 0 H -	DEC A 1 4 Z 1 H -	LD A,d8 2 8 - - - -	CCF 1 4 - 0 0 C
4x	LD B,B 1 4 - - - -	LD B,C 1 4 - - - -	LD B,D 1 4 - - - -	LD B,E 1 4 - - - -	LD B,H 1 4 - - - -	LD B,L 1 4 - - - -	LD B,(HL) 1 8 - - - -	LD B,A 1 4 - - - -	LD C,B 1 4 - - - -	LD C,C 1 4 - - - -	LD C,D 1 4 - - - -	LD C,E 1 4 - - - -	LD C,H 1 4 - - - -	LD C,L 1 4 - - - -	LD C,(HL) 1 8 - - - -	LD C,A 1 4 - - - -
5x	LD D,B 1 4 - - - -	LD D,C 1 4 - - - -	LD D,D 1 4 - - - -	LD D,E 1 4 - - - -	LD D,H 1 4 - - - -	LD D,L 1 4 - - - -	LD D,(HL) 1 8 - - - -	LD D,A 1 4 - - - -	LD E,B 1 4 - - - -	LD E,C 1 4 - - - -	LD E,D 1 4 - - - -	LD E,E 1 4 - - - -	LD E,H 1 4 - - - -	LD E,L 1 4 - - - -	LD E,(HL) 1 8 - - - -	LD E,A 1 4 - - - -
6x	LD H,B 1 4 - - - -	LD H,C 1 4 - - - -	LD H,D 1 4 - - - -	LD H,E 1 4 - - - -	LD H,H 1 4 - - - -	LD H,L 1 4 - - - -	LD H,(HL) 1 8 - - - -	LD H,A 1 4 - - - -	LD L,B 1 4 - - - -	LD L,C 1 4 - - - -	LD L,D 1 4 - - - -	LD L,E 1 4 - - - -	LD L,H 1 4 - - - -	LD L,L 1 4 - - - -	LD L,(HL) 1 8 - - - -	LD L,A 1 4 - - - -
7x	LD (HL),B 1 8 - - - -	LD (HL),C 1 8 - - - -	LD (HL),D 1 8 - - - -	LD (HL),E 1 8 - - - -	LD (HL),H 1 8 - - - -	LD (HL),L 1 8 - - - -	HALT 1 4 - - - -	LD (HL),A 1 8 - - - -	LD A,B 1 4 - - - -	LD A,C 1 4 - - - -	LD A,D 1 4 - - - -	LD A,E 1 4 - - - -	LD A,H 1 4 - - - -	LD A,L 1 4 - - - -	LD A,(HL) 1 8 - - - -	LD A,A 1 4 - - - -
8x	ADD A,B 1 4 Z 0 H C	ADD A,C 1 4 Z 0 H C	ADD A,D 1 4 Z 0 H C	ADD A,E 1 4 Z 0 H C	ADD A,H 1 4 Z 0 H C	ADD A,L 1 4 Z 0 H C	ADD A,(HL) 1 8 Z 0 H C	ADD A,A 1 4 Z 0 H C	ADC A,B 1 4 Z 0 H C	ADC A,C 1 4 Z 0 H C	ADC A,D 1 4 Z 0 H C	ADC A,E 1 4 Z 0 H C	ADC A,H 1 4 Z 0 H C	ADC A,L 1 4 Z 0 H C	ADC A,(HL) 1 8 Z 0 H C	ADC A,A 1 4 Z 0 H C
9x	SUB B 1 4 Z 1 H C	SUB C 1 4 Z 1 H C	SUB D 1 4 Z 1 H C	SUB E 1 4 Z 1 H C	SUB H 1 4 Z 1 H C	SUB L 1 4 Z 1 H C	SUB (HL) 1 8 Z 1 H C	SUB A 1 4 Z 1 H C	SBC A,B 1 4 Z 1 H C	SBC A,C 1 4 Z 1 H C	SBC A,D 1 4 Z 1 H C	SBC A,E 1 4 Z 1 H C	SBC A,H 1 4 Z 1 H C	SBC A,L 1 4 Z 1 H C	SBC A,(HL) 1 8 Z 1 H C	SBC A,A 1 4 Z 1 H C
Ax	AND B 1 4 Z 0 1 0	AND C 1 4 Z 0 1 0	AND D 1 4 Z 0 1 0	AND E 1 4 Z 0 1 0	AND H 1 4 Z 0 1 0	AND L 1 4 Z 0 1 0	AND (HL) 1 8 Z 0 1 0	AND A 1 4 Z 0 1 0	XOR B 1 4 Z 0 0 0	XOR C 1 4 Z 0 0 0	XOR D 1 4 Z 0 0 0	XOR E 1 4 Z 0 0 0	XOR H 1 4 Z 0 0 0	XOR L 1 4 Z 0 0 0	XOR (HL) 1 8 Z 0 0 0	XOR A 1 4 Z 0 0 0
Bx	OR B 1 4 Z 0 0 0	OR C 1 4 Z 0 0 0	OR D 1 4 Z 0 0 0	OR E 1 4 Z 0 0 0	OR H 1 4 Z 0 0 0	OR L 1 4 Z 0 0 0	OR (HL) 1 8 Z 0 0 0	OR A 1 4 Z 0 0 0	CP B 1 4 Z 1 H C	CP C 1 4 Z 1 H C	CP D 1 4 Z 1 H C	CP E 1 4 Z 1 H C	CP H 1 4 Z 1 H C	CP L 1 4 Z 1 H C	CP (HL) 1 8 Z 1 H C	CP A 1 4 Z 1 H C
Cx	RET NZ 1 20/8 - - - -	POP BC 1 12 - - - -	JP NZ,a16 3 16/12 - - - -	JP a16 3 16 - - - -	CALL NZ,a16 3 24/12 - - - -	PUSH BC 1 16 - - - -	ADD A,d8 2 8 Z 0 H C	RST 00H 1 16 - - - -	RET Z 1 20/8 - - - -	RET 1 16 - - - -	JP Z,a16 3 16/12 - - - -	PREFIX CB 1 4 - - - -	CALL Z,a16 3 24/12 - - - -	CALL a16 3 24 - - - -	ADC A,d8 2 8 Z 0 H C	RST 08H 1 16 - - - -
Dx	RET NC 1 20/8 - - - -	POP DE 1 12 - - - -	JP NC,a16 3 16/12 - - - -		CALL NC,a16 3 24/12 - - - -	PUSH DE 1 16 - - - -	SUB d8 2 8 Z 1 H C	RST 10H 1 16 - - - -	RET C 1 20/8 - - - -	RETI 1 16 - - - -	JP C,a16 3 16/12 - - - -		CALL C,a16 3 24/12 - - - -		SBC A,d8 2 8 Z 1 H C	RST 18H 1 16 - - - -
Ex	LDH (a8),A 2 12 - - - -	POP HL 1 12 - - - -	LD (C),A 2 8 - - - -			PUSH HL 1 16 - - - -	AND d8 2 8 Z 0 1 0	RST 20H 1 16 - - - -	ADD SP,r8 2 16 0 0 H C	JP (HL) 1 4 - - - -	LD (a16),A 3 16 - - - -				XOR d8 2 8 Z 0 0 0	RST 28H 1 16 - - - -
Fx	LDH A,(a8) 2 12 - - - -	POP AF 1 12 Z N H C	LD A,(C) 2 8 - - - -	DI 1 4 - - - -		PUSH AF 1 16 - - - -	OR d8 2 8 Z 0 0 0	RST 30H 1 16 - - - -	LD HL,SP+r8 2 12 0 0 H C	LD SP,HL 1 8 - - - -	LD A,(a16) 3 16 - - - -	EI 1 4 - - - -			CP d8 2 8 Z 1 H C	RST 38H 1 16 - - - -

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (BC),A 1 8 - - - -	INC BC 1 8 - - - -	INC B 1 4 Z 0 H -	DEC B 1 4 Z 1 H -	LD B,d8 2 8 - - - -	RLCA 1 4 0 0 0 C	LD (a16),SP 3 20 - - - -	ADD HL,BC 1 8 - 0 H C	LD A,(BC) 1 8 - - - -	DEC BC 1 8 - - - -	INC C 1 4 Z 0 H -	DEC C 1 4 Z 1 H -	LD C,d8 2 8 - - - -	RRCA 1 4 0 0 0 C
1x	STOP 0 2 4 - - - -	LD DE,d16 3 12 - - - -	LD (DE),A 1 8 - - - -	INC DE 1 8 - - - -	INC D 1 4 Z 0 H -	DEC D 1 4 Z 1 H -	LD D,d8 2 8 - - - -	RLA 1 4 0 0 0 C	JR r8 2 12 - - - -	ADD HL,DE 1 8 - 0 H C	LD A,(DE) 1 8 - - - -	DEC DE 1 8 - - - -	INC E 1 4 Z 0 H -	DEC E 1 4 Z 1 H -	LD E,d8 2 8 - - - -	RRA 1 4 0 0 0 C
2x	JR NZ,r8 2 12/8 - - - -	LD HL,d16 3 12 - - - -	LD (HL+),A 1 8 - - - -	INC HL 1 8 - - - -	INC H 1 4 Z 0 H -	DEC H 1 4 Z 1 H -	LD H,d8 2 8 - - - -	DAA 1 4 Z - 0 C	JR Z,r8 2 12/8 - - - -	ADD HL,HL 1 8 - 0 H C	LD A,(HL+) 1 8 - - - -	DEC HL 1 8 - - - -	INC L 1 4 Z 0 H -	DEC L 1 4 Z 1 H -	LD L,d8 2 8 - - - -	CPL 1 4 - 1 1 -
3x	JR NC,r8 2 12/8 - - - -	LD SP,d16 3 12 - - - -	LD (HL-),A 1 8 - - - -	INC SP 1 8 - - - -	INC (HL) 1 12 Z 0 H -	DEC (HL) 1 12 Z 1 H -	LD (HL),d8 2 12 - - - -	SCF 1 4 - 0 0 1	JR C,r8 2 12/8 - - - -	ADD HL,SP 1 8 - 0 H C	LD A,(HL-) 1 8 - - - -	DEC SP 1 8 - - - -	INC A 1 4 Z 0 H -	DEC A 1 4 Z 1 H -	LD A,d8 2 8 - - - -	CCF 1 4 - 0 0 C
4x	LD B,B 1 4 - - - -	LD B,C 1 4 - - - -	LD B,D 1 4 - - - -	LD B,E 1 4 - - - -	LD B,H 1 4 - - - -	LD B,L 1 4 - - - -	LD B,(HL) 1 8 - - - -	LD B,A 1 4 - - - -	LD C,B 1 4 - - - -	LD C,C 1 4 - - - -	LD C,D 1 4 - - - -	LD C,E 1 4 - - - -	LD C,H 1 4 - - - -	LD C,L 1 4 - - - -	LD C,(HL) 1 8 - - - -	LD C,A 1 4 - - - -
5x	LD D,B 1 4 - - - -	LD D,C 1 4 - - - -	LD D,D 1 4 - - - -	LD D,E 1 4 - - - -	LD D,H 1 4 - - - -	LD D,L 1 4 - - - -	LD D,(HL) 1 8 - - - -	LD D,A 1 4 - - - -	LD E,B 1 4 - - - -	LD E,C 1 4 - - - -	LD E,D 1 4 - - - -	LD E,E 1 4 - - - -	LD E,H 1 4 - - - -	LD E,L 1 4 - - - -	LD E,(HL) 1 8 - - - -	LD E,A 1 4 - - - -
6x	LD H,B 1 4 - - - -	LD H,C 1 4 - - - -	LD H,D 1 4 - - - -	LD H,E 1 4 - - - -	LD H,H 1 4 - - - -	LD H,L 1 4 - - - -	LD H,(HL) 1 8 - - - -	LD H,A 1 4 - - - -	LD L,B 1 4 - - - -	LD L,C 1 4 - - - -	LD L,D 1 4 - - - -	LD L,E 1 4 - - - -	LD L,H 1 4 - - - -	LD L,L 1 4 - - - -	LD L,(HL) 1 8 - - - -	LD L,A 1 4 - - - -
7x	LD (HL),B 1 8 - - - -	LD (HL),C 1 8 - - - -	LD (HL),D 1 8 - - - -	LD (HL),E 1 8 - - - -	LD (HL),H 1 8 - - - -	LD (HL),L 1 8 - - - -	HALT 1 4 - - - -	LD (HL),A 1 8 - - - -	LD A,B 1 4 - - - -	LD A,C 1 4 - - - -	LD A,D 1 4 - - - -	LD A,E 1 4 - - - -	LD A,H 1 4 - - - -	LD A,L 1 4 - - - -	LD A,(HL) 1 8 - - - -	LD A,A 1 4 - - - -
8x	ADD A,B 1 4 Z 0 H C	ADD A,C 1 4 Z 0 H C	ADD A,D 1 4 Z 0 H C	ADD A,E 1 4 Z 0 H C	ADD A,H 1 4 Z 0 H C	ADD A,L 1 4 Z 0 H C	ADD A,(HL) 1 8 Z 0 H C	ADD A,A 1 4 Z 0 H C	ADC A,B 1 4 Z 0 H C	ADC A,C 1 4 Z 0 H C	ADC A,D 1 4 Z 0 H C	ADC A,E 1 4 Z 0 H C	ADC A,H 1 4 Z 0 H C	ADC A,L 1 4 Z 0 H C	ADC A,(HL) 1 8 Z 0 H C	ADC A,A 1 4 Z 0 H C
9x	SUB B 1 4 Z 1 H C	SUB C 1 4 Z 1 H C	SUB D 1 4 Z 1 H C	SUB E 1 4 Z 1 H C	SUB H 1 4 Z 1 H C	SUB L 1 4 Z 1 H C	SUB (HL) 1 8 Z 1 H C	SUB A 1 4 Z 1 H C	SBC A,B 1 4 Z 1 H C	SBC A,C 1 4 Z 1 H C	SBC A,D 1 4 Z 1 H C	SBC A,E 1 4 Z 1 H C	SBC A,H 1 4 Z 1 H C	SBC A,L 1 4 Z 1 H C	SBC A,(HL) 1 8 Z 1 H C	SBC A,A 1 4 Z 1 H C
Ax	AND B 1 4 Z 0 1 0	AND C 1 4 Z 0 1 0	AND D 1 4 Z 0 1 0	AND E 1 4 Z 0 1 0	AND H 1 4 Z 0 1 0	AND L 1 4 Z 0 1 0	AND (HL) 1 8 Z 0 1 0	AND A 1 4 Z 0 1 0	XOR B 1 4 Z 0 0 0	XOR C 1 4 Z 0 0 0	XOR D 1 4 Z 0 0 0	XOR E 1 4 Z 0 0 0	XOR H 1 4 Z 0 0 0	XOR L 1 4 Z 0 0 0	XOR (HL) 1 8 Z 0 0 0	XOR A 1 4 Z 0 0 0
Bx	OR B 1 4 Z 0 0 0	OR C 1 4 Z 0 0 0	OR D 1 4 Z 0 0 0	OR E 1 4 Z 0 0 0	OR H 1 4 Z 0 0 0	OR L 1 4 Z 0 0 0	OR (HL) 1 8 Z 0 0 0	OR A 1 4 Z 0 0 0	CP B 1 4 Z 1 H C	CP C 1 4 Z 1 H C	CP D 1 4 Z 1 H C	CP E 1 4 Z 1 H C	CP H 1 4 Z 1 H C	CP L 1 4 Z 1 H C	CP (HL) 1 8 Z 1 H C	CP A 1 4 Z 1 H C
Cx	RET NZ 1 20/8 - - - -	POP BC 1 12 - - - -	JP NZ,a16 3 16/12 - - - -	JP a16 3 16 - - - -	CALL NZ,a16 3 24/12 - - - -	PUSH BC 1 16 - - - -	ADD A,d8 2 8 Z 0 H C	RST 00H 1 16 - - - -	RET Z 1 20/8 - - - -	RET 1 16 - - - -	JP Z,a16 3 16/12 - - - -	PREFIX CB 1 4 - - - -	CALL Z,a16 3 24/12 - - - -	CALL a16 3 24 - - - -	ADC A,d8 2 8 Z 0 H C	RST 08H 1 16 - - - -
Dx	RET NC 1 20/8 - - - -	POP DE 1 12 - - - -	JP NC,a16 3 16/12 - - - -		CALL NC,a16 3 24/12 - - - -	PUSH DE 1 16 - - - -	SUB d8 2 8 Z 1 H C	RST 10H 1 16 - - - -	RET C 1 20/8 - - - -	RETI 1 16 - - - -	JP C,a16 3 16/12 - - - -		CALL C,a16 3 24/12 - - - -		SBC A,d8 2 8 Z 1 H C	RST 18H 1 16 - - - -
Ex	LDH (a8),A 2 12 - - - -	POP HL 1 12 - - - -	LD (C),A 2 8 - - - -			PUSH HL 1 16 - - - -	AND d8 2 8 Z 0 1 0	RST 20H 1 16 - - - -	ADD SP,r8 2 16 0 0 H C	JP (HL) 1 4 - - - -	LD (a16),A 3 16 - - - -				XOR d8 2 8 Z 0 0 0	RST 28H 1 16 - - - -
Fx	LDH A,(a8) 2 12 - - - -	POP AF 1 12 Z N H C	LD A,(C) 2 8 - - - -	DI 1 4 - - - -		PUSH AF 1 16 - - - -	OR d8 2 8 Z 0 0 0	RST 30H 1 16 - - - -	LD HL,SP+r8 2 12 0 0 H C	LD SP,HL 1 8 - - - -	LD A,(a16) 3 16 - - - -	EI 1 4 - - - -			CP d8 2 8 Z 1 H C	RST 38H 1 16 - - - -

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (BC),A 1 8 - - - -	INC BC 1 8 - - - -	INC B 1 4 Z 0 H -	DEC B 1 4 Z 1 H -	LD B,d8 2 8 - - - -	RLCA 1 4 0 0 0 C	LD (a16),SP 3 20 - - - -	ADD HL,BC 1 8 - 0 H C	LD A,(BC) 1 8 - - - -	DEC BC 1 8 - - - -	INC C 1 4 Z 0 H -	DEC C 1 4 Z 1 H -	LD C,d8 2 8 - - - -	RRCA 1 4 0 0 0 C
1x	STOP 0 2 4 - - - -	LD DE,d16 3 12 - - - -	LD (DE),A 1 8 - - - -	INC DE 1 8 - - - -	INC D 1 4 Z 0 H -	DEC D 1 4 Z 1 H -	LD D,d8 2 8 - - - -	RLA 1 4 0 0 0 C	JR r8 2 12 - - - -	ADD HL,DE 1 8 - 0 H C	LD A,(DE) 1 8 - - - -	DEC DE 1 8 - - - -	INC E 1 4 Z 0 H -	DEC E 1 4 Z 1 H -	LD E,d8 2 8 - - - -	RRA 1 4 0 0 0 C
2x	JR NZ,r8 2 12/8 - - - -	LD HL,d16 3 12 - - - -	LD (HL+),A 1 8 - - - -	INC HL 1 8 - - - -	INC H 1 4 Z 0 H -	DEC H 1 4 Z 1 H -	LD H,d8 2 8 - - - -	DAA 1 4 Z - 0 C	JR Z,r8 2 12/8 - - - -	ADD HL,HL 1 8 - 0 H C	LD A,(HL+) 1 8 - - - -	DEC HL 1 8 - - - -	INC L 1 4 Z 0 H -	DEC L 1 4 Z 1 H -	LD L,d8 2 8 - - - -	CPL 1 4 - 1 1 -
3x	JR NC,r8 2 12/8 - - - -	LD SP,d16 3 12 - - - -	LD (HL-), 1 8 - - - -											DEC A 1 4 Z 1 H -	LD A,d8 2 8 - - - -	CCF 1 4 - 0 0 C
4x	LD B,B 1 4 - - - -	LD B,C 1 4 - - - -	LD B,D 1 4 - - - -											LD C,L 1 4 - - - -	LD C,(HL) 1 8 - - - -	LD C,A 1 4 - - - -
5x	LD D,B 1 4 - - - -	LD D,C 1 4 - - - -	LD D,D 1 4 - - - -											LD E,L 1 4 - - - -	LD E,(HL) 1 8 - - - -	LD E,A 1 4 - - - -
6x	LD H,B 1 4 - - - -	LD H,C 1 4 - - - -	LD H,D 1 4 - - - -											LD L,L 1 4 - - - -	LD L,(HL) 1 8 - - - -	LD L,A 1 4 - - - -
7x	LD (HL),B 1 8 - - - -	LD (HL),C 1 8 - - - -	LD (HL), 1 8 - - - -											LD A,L 1 4 - - - -	LD A,(HL) 1 8 - - - -	LD A,A 1 4 - - - -
8x	ADD A,B 1 4 Z 0 H C	ADD A,C 1 4 Z 0 H C	ADD A,D 1 4 Z 0 H C											ADC A,L 1 4 Z 0 H C	ADC A,(HL) 1 8 Z 0 H C	ADC A,A 1 4 Z 0 H C
9x	SUB B 1 4 Z 1 H C	SUB C 1 4 Z 1 H C	SUB D 1 4 Z 1 H C											SBC A,L 1 4 Z 1 H C	SBC A,(HL) 1 8 Z 1 H C	SBC A,A 1 4 Z 1 H C
Ax	AND B 1 4 Z 0 1 0	AND C 1 4 Z 0 1 0	AND D 1 4 Z 0 1 0											XOR L 1 4 Z 0 0 0	XOR (HL) 1 8 Z 0 0 0	XOR A 1 4 Z 0 0 0
Bx	OR B 1 4 Z 0 0 0	OR C 1 4 Z 0 0 0	OR D 1 4 Z 0 0 0											CP L 1 4 Z 1 H C	CP (HL) 1 8 Z 1 H C	CP A 1 4 Z 1 H C
Cx	RET NZ 1 20/8 - - - -	POP BC 1 12 - - - -	JP NZ,a1 3 16/12 - - - -										6	CALL a16 3 24 - - - -	ADC A,d8 2 8 Z 0 H C	RST 08H 1 16 - - - -
Dx	RET NC 1 20/8 - - - -	POP DE 1 12 - - - -	JP NC,a16 3 16/12 - - - -		CALL NC,a16 3 24/12 - - - -	PUSH DE 1 16 - - - -	SUB d8 2 8 Z 1 H C	RST 10H 1 16 - - - -	RET C 1 20/8 - - - -	RETI 1 16 - - - -	JP C,a16 3 16/12 - - - -		CALL C,a16 3 24/12 - - - -		SBC A,d8 2 8 Z 1 H C	RST 18H 1 16 - - - -
Ex	LDH (a8),A 2 12 - - - -	POP HL 1 12 - - - -	LD (C),A 2 8 - - - -			PUSH HL 1 16 - - - -	AND d8 2 8 Z 0 1 0	RST 20H 1 16 - - - -	ADD SP,r8 2 16 0 0 H C	JP (HL) 1 4 - - - -	LD (a16),A 3 16 - - - -				XOR d8 2 8 Z 0 0 0	RST 28H 1 16 - - - -
Fx	LDH A,(a8) 2 12 - - - -	POP AF 1 12 Z N H C	LD A,(C) 2 8 - - - -	DI 1 4 - - - -		PUSH AF 1 16 - - - -	OR d8 2 8 Z 0 0 0	RST 30H 1 16 - - - -	LD HL,SP+r8 2 12 0 0 H C	LD SP,HL 1 8 - - - -	LD A,(a16) 3 16 - - - -	EI 1 4 - - - -			CP d8 2 8 Z 1 H C	RST 38H 1 16 - - - -

PREFIX CB

1 4

- - - -

CB

	x0	x1	x2	x3	x4	x5	x6	x7
0x	RLC B 2 8 Z 0 0 C	RLC C 2 8 Z 0 0 C	RLC D 2 8 Z 0 0 C	RLC E 2 8 Z 0 0 C	RLC H 2 8 Z 0 0 C	RLC L 2 8 Z 0 0 C	RLC (HL) 2 16 Z 0 0 C	RLC A 2 8 Z 0 0 C
1x	RL B 2 8 Z 0 0 C	RL C 2 8 Z 0 0 C	RL D 2 8 Z 0 0 C	RL E 2 8 Z 0 0 C	RL H 2 8 Z 0 0 C	RL L 2 8 Z 0 0 C	RL (HL) 2 16 Z 0 0 C	RL A 2 8 Z 0 0 C
2x	SLA B 2 8 Z 0 0 C	SLA C 2 8 Z 0 0 C	SLA D 2 8 Z 0 0 C	SLA E 2 8 Z 0 0 C	SLA H 2 8 Z 0 0 C	SLA L 2 8 Z 0 0 C	SLA (HL) 2 16 Z 0 0 C	SLA A 2 8 Z 0 0 C
3x	SWAP B 2 8 Z 0 0 0	SWAP C 2 8 Z 0 0 0	SWAP D 2 8 Z 0 0 0	SWAP E 2 8 Z 0 0 0	SWAP H 2 8 Z 0 0 0	SWAP L 2 8 Z 0 0 0	SWAP (HL) 2 16 Z 0 0 0	SWAP A 2 8 Z 0 0 0
4x	BIT 0,B 2 8 Z 0 1 -	BIT 0,C 2 8 Z 0 1 -	BIT 0,D 2 8 Z 0 1 -	BIT 0,E 2 8 Z 0 1 -	BIT 0,H 2 8 Z 0 1 -	BIT 0,L 2 8 Z 0 1 -	BIT 0,(HL) 2 16 Z 0 1 -	BIT 0,A 2 8 Z 0 1 -
5x	BIT 2,B 2 8 Z 0 1 -	BIT 2,C 2 8 Z 0 1 -	BIT 2,D 2 8 Z 0 1 -	BIT 2,E 2 8 Z 0 1 -	BIT 2,H 2 8 Z 0 1 -	BIT 2,L 2 8 Z 0 1 -	BIT 2,(HL) 2 16 Z 0 1 -	BIT 2,A 2 8 Z 0 1 -
6x	BIT 4,B 2 8 Z 0 1 -	BIT 4,C 2 8 Z 0 1 -	BIT 4,D 2 8 Z 0 1 -	BIT 4,E 2 8 Z 0 1 -	BIT 4,H 2 8 Z 0 1 -	BIT 4,L 2 8 Z 0 1 -	BIT 4,(HL) 2 16 Z 0 1 -	BIT 4,A 2 8 Z 0 1 -
7x	BIT 6,B 2 8 Z 0 1 -	BIT 6,C 2 8 Z 0 1 -	BIT 6,D 2 8 Z 0 1 -	BIT 6,E 2 8 Z 0 1 -	BIT 6,H 2 8 Z 0 1 -	BIT 6,L 2 8 Z 0 1 -	BIT 6,(HL) 2 16 Z 0 1 -	BIT 6,A 2 8 Z 0 1 -
8x	RES 0,B 2 8 - - - -	RES 0,C 2 8 - - - -	RES 0,D 2 8 - - - -	RES 0,E 2 8 - - - -	RES 0,H 2 8 - - - -	RES 0,L 2 8 - - - -	RES 0,(HL) 2 16 - - - -	RES 0,A 2 8 - - - -
9x	RES 2,B 2 8 - - - -	RES 2,C 2 8 - - - -	RES 2,D 2 8 - - - -	RES 2,E 2 8 - - - -	RES 2,H 2 8 - - - -	RES 2,L 2 8 - - - -	RES 2,(HL) 2 16 - - - -	RES 2,A 2 8 - - - -
Ax	RES 4,B 2 8 - - - -	RES 4,C 2 8 - - - -	RES 4,D 2 8 - - - -	RES 4,E 2 8 - - - -	RES 4,H 2 8 - - - -	RES 4,L 2 8 - - - -	RES 4,(HL) 2 16 - - - -	RES 4,A 2 8 - - - -
Bx	RES 6,B 2 8 - - - -	RES 6,C 2 8 - - - -	RES 6,D 2 8 - - - -	RES 6,E 2 8 - - - -	RES 6,H 2 8 - - - -	RES 6,L 2 8 - - - -	RES 6,(HL) 2 16 - - - -	RES 6,A 2 8 - - - -
Cx	SET 0,B 2 8 - - - -	SET 0,C 2 8 - - - -	SET 0,D 2 8 - - - -	SET 0,E 2 8 - - - -	SET 0,H 2 8 - - - -	SET 0,L 2 8 - - - -	SET 0,(HL) 2 16 - - - -	SET 0,A 2 8 - - - -
Dx	SET 2,B 2 8 - - - -	SET 2,C 2 8 - - - -	SET 2,D 2 8 - - - -	SET 2,E 2 8 - - - -	SET 2,H 2 8 - - - -	SET 2,L 2 8 - - - -	SET 2,(HL) 2 16 - - - -	SET 2,A 2 8 - - - -
Ex	SET 4,B 2 8 - - - -	SET 4,C 2 8 - - - -	SET 4,D 2 8 - - - -	SET 4,E 2 8 - - - -	SET 4,H 2 8 - - - -	SET 4,L 2 8 - - - -	SET 4,(HL) 2 16 - - - -	SET 4,A 2 8 - - - -
Fx	SET 6,B 2 8 - - - -	SET 6,C 2 8 - - - -	SET 6,D 2 8 - - - -	SET 6,E 2 8 - - - -	SET 6,H 2 8 - - - -	SET 6,L 2 8 - - - -	SET 6,(HL) 2 16 - - - -	SET 6,A 2 8 - - - -

	x8	x9	xA	xB	xC	xD	xE	xF
0x	RRC B 2 8 Z 0 0 C	RRC C 2 8 Z 0 0 C	RRC D 2 8 Z 0 0 C	RRC E 2 8 Z 0 0 C	RRC H 2 8 Z 0 0 C	RRC L 2 8 Z 0 0 C	RRC (HL) 2 16 Z 0 0 C	RRC A 2 8 Z 0 0 C
1x	RR B 2 8 Z 0 0 C	RR C 2 8 Z 0 0 C	RR D 2 8 Z 0 0 C	RR E 2 8 Z 0 0 C	RR H 2 8 Z 0 0 C	RR L 2 8 Z 0 0 C	RR (HL) 2 16 Z 0 0 C	RR A 2 8 Z 0 0 C
2x	SRA B 2 8 Z 0 0 0	SRA C 2 8 Z 0 0 0	SRA D 2 8 Z 0 0 0	SRA E 2 8 Z 0 0 0	SRA H 2 8 Z 0 0 0	SRA L 2 8 Z 0 0 0	SRA (HL) 2 16 Z 0 0 0	SRA A 2 8 Z 0 0 0
3x	SRL B 2 8 Z 0 0 C	SRL C 2 8 Z 0 0 C	SRL D 2 8 Z 0 0 C	SRL E 2 8 Z 0 0 C	SRL H 2 8 Z 0 0 C	SRL L 2 8 Z 0 0 C	SRL (HL) 2 16 Z 0 0 C	SRL A 2 8 Z 0 0 C
4x	BIT 1,B 2 8 Z 0 1 -	BIT 1,C 2 8 Z 0 1 -	BIT 1,D 2 8 Z 0 1 -	BIT 1,E 2 8 Z 0 1 -	BIT 1,H 2 8 Z 0 1 -	BIT 1,L 2 8 Z 0 1 -	BIT 1,(HL) 2 16 Z 0 1 -	BIT 1,A 2 8 Z 0 1 -
5x	BIT 3,B 2 8 Z 0 1 -	BIT 3,C 2 8 Z 0 1 -	BIT 3,D 2 8 Z 0 1 -	BIT 3,E 2 8 Z 0 1 -	BIT 3,H 2 8 Z 0 1 -	BIT 3,L 2 8 Z 0 1 -	BIT 3,(HL) 2 16 Z 0 1 -	BIT 3,A 2 8 Z 0 1 -
6x	BIT 5,B 2 8 Z 0 1 -	BIT 5,C 2 8 Z 0 1 -	BIT 5,D 2 8 Z 0 1 -	BIT 5,E 2 8 Z 0 1 -	BIT 5,H 2 8 Z 0 1 -	BIT 5,L 2 8 Z 0 1 -	BIT 5,(HL) 2 16 Z 0 1 -	BIT 5,A 2 8 Z 0 1 -
7x	BIT 7,B 2 8 Z 0 1 -	BIT 7,C 2 8 Z 0 1 -	BIT 7,D 2 8 Z 0 1 -	BIT 7,E 2 8 Z 0 1 -	BIT 7,H 2 8 Z 0 1 -	BIT 7,L 2 8 Z 0 1 -	BIT 7,(HL) 2 16 Z 0 1 -	BIT 7,A 2 8 Z 0 1 -
8x	RES 1,B 2 8 - - - -	RES 1,C 2 8 - - - -	RES 1,D 2 8 - - - -	RES 1,E 2 8 - - - -	RES 1,H 2 8 - - - -	RES 1,L 2 8 - - - -	RES 1,(HL) 2 16 - - - -	RES 1,A 2 8 - - - -
9x	RES 3,B 2 8 - - - -	RES 3,C 2 8 - - - -	RES 3,D 2 8 - - - -	RES 3,E 2 8 - - - -	RES 3,H 2 8 - - - -	RES 3,L 2 8 - - - -	RES 3,(HL) 2 16 - - - -	RES 3,A 2 8 - - - -
Ax	RES 5,B 2 8 - - - -	RES 5,C 2 8 - - - -	RES 5,D 2 8 - - - -	RES 5,E 2 8 - - - -	RES 5,H 2 8 - - - -	RES 5,L 2 8 - - - -	RES 5,(HL) 2 16 - - - -	RES 5,A 2 8 - - - -
Bx	RES 7,B 2 8 - - - -	RES 7,C 2 8 - - - -	RES 7,D 2 8 - - - -	RES 7,E 2 8 - - - -	RES 7,H 2 8 - - - -	RES 7,L 2 8 - - - -	RES 7,(HL) 2 16 - - - -	RES 7,A 2 8 - - - -
Cx	SET 1,B 2 8 - - - -	SET 1,C 2 8 - - - -	SET 1,D 2 8 - - - -	SET 1,E 2 8 - - - -	SET 1,H 2 8 - - - -	SET 1,L 2 8 - - - -	SET 1,(HL) 2 16 - - - -	SET 1,A 2 8 - - - -
Dx	SET 3,B 2 8 - - - -	SET 3,C 2 8 - - - -	SET 3,D 2 8 - - - -	SET 3,E 2 8 - - - -	SET 3,H 2 8 - - - -	SET 3,L 2 8 - - - -	SET 3,(HL) 2 16 - - - -	SET 3,A 2 8 - - - -
Ex	SET 5,B 2 8 - - - -	SET 5,C 2 8 - - - -	SET 5,D 2 8 - - - -	SET 5,E 2 8 - - - -	SET 5,H 2 8 - - - -	SET 5,L 2 8 - - - -	SET 5,(HL) 2 16 - - - -	SET 5,A 2 8 - - - -
Fx	SET 7,B 2 8 - - - -	SET 7,C 2 8 - - - -	SET 7,D 2 8 - - - -	SET 7,E 2 8 - - - -	SET 7,H 2 8 - - - -	SET 7,L 2 8 - - - -	SET 7,(HL) 2 16 - - - -	SET 7,A 2 8 - - - -

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (BC),A 1 8 - - - -	INC BC 1 8 - - - -	INC B 1 4 Z 0 H -	DEC B 1 4 Z 1 H -	LD B,d8 2 8 - - - -	RLCA 1 4 0 0 0 C	LD (a16),SP 3 20 - - - -	ADD HL,BC 1 8 - 0 H C	LD A,(BC) 1 8 - - - -	DEC BC 1 8 - - - -	INC C 1 4 Z 0 H -	DEC C 1 4 Z 1 H -	LD C,d8 2 8 - - - -	RRCA 1 4 0 0 0 C
1x	STOP 0 2 4 - - - -	LD DE,d16 3 12 - - - -	LD (DE),A 1 8 - - - -	INC DE 1 8 - - - -	INC D 1 4 Z 0 H -	DEC D 1 4 Z 1 H -	LD D,d8 2 8 - - - -	RLA 1 4 0 0 0 C	JR r8 2 12 - - - -	ADD HL,DE 1 8 - 0 H C	LD A,(DE) 1 8 - - - -	DEC DE 1 8 - - - -	INC E 1 4 Z 0 H -	DEC E 1 4 Z 1 H -	LD E,d8 2 8 - - - -	RRA 1 4 0 0 0 C
2x	JR NZ,r8 2 12/8 - - - -	LD HL,d16 3 12 - - - -	LD (HL+),A 1 8 - - - -	INC HL 1 8 - - - -	INC H 1 4 Z 0 H -	DEC H 1 4 Z 1 H -	LD H,d8 2 8 - - - -	DAA 1 4 Z - 0 C	JR Z,r8 2 12/8 - - - -	ADD HL,HL 1 8 - 0 H C	LD A,(HL+) 1 8 - - - -	DEC HL 1 8 - - - -	INC L 1 4 Z 0 H -	DEC L 1 4 Z 1 H -	LD L,d8 2 8 - - - -	CPL 1 4 - 1 1 -
3x	JR NC,r8 2 12/8 - - - -	LD SP,d16 3 12 - - - -	LD (HL-),A 1 8 - - - -	INC SP 1 8 - - - -	INC (HL) 1 12 Z 0 H -	DEC (HL) 1 12 Z 1 H -	LD (HL),d8 2 12 - - - -	SCF 1 4 - 0 0 1	JR C,r8 2 12/8 - - - -	ADD HL,SP 1 8 - 0 H C	LD A,(HL-) 1 8 - - - -	DEC SP 1 8 - - - -	INC A 1 4 Z 0 H -	DEC A 1 4 Z 1 H -	LD A,d8 2 8 - - - -	CCF 1 4 - 0 0 C
4x	LD B,B 1 4 - - - -	LD B,C 1 4 - - - -	LD B,D 1 4 - - - -	LD B,E 1 4 - - - -	LD B,H 1 4 - - - -	LD B,L 1 4 - - - -	LD B,(HL) 1 8 - - - -	LD B,A 1 4 - - - -	LD C,B 1 4 - - - -	LD C,C 1 4 - - - -	LD C,D 1 4 - - - -	LD C,E 1 4 - - - -	LD C,H 1 4 - - - -	LD C,L 1 4 - - - -	LD C,(HL) 1 8 - - - -	LD C,A 1 4 - - - -
5x	LD D,B 1 4 - - - -	LD D,C 1 4 - - - -	LD D,D 1 4 - - - -	LD D,E 1 4 - - - -	LD D,H 1 4 - - - -	LD D,L 1 4 - - - -	LD D,(HL) 1 8 - - - -	LD D,A 1 4 - - - -	LD E,B 1 4 - - - -	LD E,C 1 4 - - - -	LD E,D 1 4 - - - -	LD E,E 1 4 - - - -	LD E,H 1 4 - - - -	LD E,L 1 4 - - - -	LD E,(HL) 1 8 - - - -	LD E,A 1 4 - - - -
6x	LD H,B 1 4 - - - -	LD H,C 1 4 - - - -	LD H,D 1 4 - - - -	LD H,E 1 4 - - - -	LD H,H 1 4 - - - -	LD H,L 1 4 - - - -	LD H,(HL) 1 8 - - - -	LD H,A 1 4 - - - -	LD L,B 1 4 - - - -	LD L,C 1 4 - - - -	LD L,D 1 4 - - - -	LD L,E 1 4 - - - -	LD L,H 1 4 - - - -	LD L,L 1 4 - - - -	LD L,(HL) 1 8 - - - -	LD L,A 1 4 - - - -
7x	LD (HL),B 1 8 - - - -	LD (HL),C 1 8 - - - -	LD (HL),D 1 8 - - - -	LD (HL),E 1 8 - - - -	LD (HL),H 1 8 - - - -	LD (HL),L 1 8 - - - -	HALT 1 4 - - - -	LD (HL),A 1 8 - - - -	LD A,B 1 4 - - - -	LD A,C 1 4 - - - -	LD A,D 1 4 - - - -	LD A,E 1 4 - - - -	LD A,H 1 4 - - - -	LD A,L 1 4 - - - -	LD A,(HL) 1 8 - - - -	LD A,A 1 4 - - - -
8x	ADD A,B 1 4 Z 0 H C	ADD A,C 1 4 Z 0 H C	ADD A,D 1 4 Z 0 H C	ADD A,E 1 4 Z 0 H C	ADD A,H 1 4 Z 0 H C	ADD A,L 1 4 Z 0 H C	ADD A,(HL) 1 8 Z 0 H C	ADD A,A 1 4 Z 0 H C	ADC A,B 1 4 Z 0 H C	ADC A,C 1 4 Z 0 H C	ADC A,D 1 4 Z 0 H C	ADC A,E 1 4 Z 0 H C	ADC A,H 1 4 Z 0 H C	ADC A,L 1 4 Z 0 H C	ADC A,(HL) 1 8 Z 0 H C	ADC A,A 1 4 Z 0 H C
9x	SUB B 1 4 Z 1 H C	SUB C 1 4 Z 1 H C	SUB D 1 4 Z 1 H C	SUB E 1 4 Z 1 H C	SUB H 1 4 Z 1 H C	SUB L 1 4 Z 1 H C	SUB (HL) 1 8 Z 1 H C	SUB A 1 4 Z 1 H C	SBC A,B 1 4 Z 1 H C	SBC A,C 1 4 Z 1 H C	SBC A,D 1 4 Z 1 H C	SBC A,E 1 4 Z 1 H C	SBC A,H 1 4 Z 1 H C	SBC A,L 1 4 Z 1 H C	SBC A,(HL) 1 8 Z 1 H C	SBC A,A 1 4 Z 1 H C
Ax	AND B 1 4 Z 0 1 0	AND C 1 4 Z 0 1 0	AND D 1 4 Z 0 1 0	AND E 1 4 Z 0 1 0	AND H 1 4 Z 0 1 0	AND L 1 4 Z 0 1 0	AND (HL) 1 8 Z 0 1 0	AND A 1 4 Z 0 1 0	XOR B 1 4 Z 0 0 0	XOR C 1 4 Z 0 0 0	XOR D 1 4 Z 0 0 0	XOR E 1 4 Z 0 0 0	XOR H 1 4 Z 0 0 0	XOR L 1 4 Z 0 0 0	XOR (HL) 1 8 Z 0 0 0	XOR A 1 4 Z 0 0 0
Bx	OR B 1 4 Z 0 0 0	OR C 1 4 Z 0 0 0	OR D 1 4 Z 0 0 0	OR E 1 4 Z 0 0 0	OR H 1 4 Z 0 0 0	OR L 1 4 Z 0 0 0	OR (HL) 1 8 Z 0 0 0	OR A 1 4 Z 0 0 0	CP B 1 4 Z 1 H C	CP C 1 4 Z 1 H C	CP D 1 4 Z 1 H C	CP E 1 4 Z 1 H C	CP H 1 4 Z 1 H C	CP L 1 4 Z 1 H C	CP (HL) 1 8 Z 1 H C	CP A 1 4 Z 1 H C
Cx	RET NZ 1 20/8 - - - -	POP BC 1 12 - - - -	JP NZ,a16 3 16/12 - - - -	JP a16 3 16 - - - -	CALL NZ,a16 3 24/12 - - - -	PUSH BC 1 16 - - - -	ADD A,d8 2 8 Z 0 H C	RST 00H 1 16 - - - -	RET Z 1 20/8 - - - -	RET 1 16 - - - -	JP Z,a16 3 16/12 - - - -	PREFIX CB 1 4 - - - -	CALL Z,a16 3 24/12 - - - -	CALL a16 3 24 - - - -	ADC A,d8 2 8 Z 0 H C	RST 08H 1 16 - - - -
Dx	RET NC 1 20/8 - - - -	POP DE 1 12 - - - -	JP NC,a16 3 16/12 - - - -		CALL NC,a16 3 24/12 - - - -	PUSH DE 1 16 - - - -	SUB d8 2 8 Z 1 H C	RST 10H 1 16 - - - -	RET C 1 20/8 - - - -	RETI 1 16 - - - -	JP C,a16 3 16/12 - - - -		CALL C,a16 3 24/12 - - - -		SBC A,d8 2 8 Z 1 H C	RST 18H 1 16 - - - -
Ex	LDH (a8),A 2 12 - - - -	POP HL 1 12 - - - -	LD (C),A 2 8 - - - -			PUSH HL 1 16 - - - -	AND d8 2 8 Z 0 1 0	RST 20H 1 16 - - - -	ADD SP,r8 2 16 0 0 H C	JP (HL) 1 4 - - - -	LD (a16),A 3 16 - - - -				XOR d8 2 8 Z 0 0 0	RST 28H 1 16 - - - -
Fx	LDH A,(a8) 2 12 - - - -	POP AF 1 12 Z N H C	LD A,(C) 2 8 - - - -	DI 1 4 - - - -		PUSH AF 1 16 - - - -	OR d8 2 8 Z 0 0 0	RST 30H 1 16 - - - -	LD HL,SP+r8 2 12 0 0 H C	LD SP,HL 1 8 - - - -	LD A,(a16) 3 16 - - - -	EI 1 4 - - - -			CP d8 2 8 Z 1 H C	RST 38H 1 16 - - - -

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (BC),A 1 8 - - - -	INC BC 1 8 - - - -	INC B 1 4 Z 0 H -	DEC B 1 4 Z 1 H -	LD B,d8 2 8 - - - -	RLCA 1 4 0 0 0 C	LD (a16),SP 3 20 - - - -	ADD HL,BC 1 8 - 0 H C	LD A,(BC) 1 8 - - - -	DEC BC 1 8 - - - -	INC C 1 4 Z 0 H -	DEC C 1 4 Z 1 H -	LD C,d8 2 8 - - - -	RRCA 1 4 0 0 0 C
1x	STOP 0 2 4 - - - -	LD DE,d16 3 12 - - - -	LD (DE),A 1 8 - - - -	INC DE 1 8 - - - -	INC D 1 4 Z 0 H -	DEC D 1 4 Z 1 H -	LD D,d8 2 8 - - - -	RLA 1 4 0 0 0 C	JR r8 2 12 - - - -	ADD HL,DE 1 8 - 0 H C	LD A,(DE) 1 8 - - - -	DEC DE 1 8 - - - -	INC E 1 4 Z 0 H -	DEC E 1 4 Z 1 H -	LD E,d8 2 8 - - - -	RRA 1 4 0 0 0 C
2x	JR NZ,r8 2 12/8 - - - -	LD HL,d16 3 12 - - - -	LD (HL+),A 1 8 - - - -	INC HL 1 8 - - - -	INC H 1 4 Z 0 H -	DEC H 1 4 Z 1 H -	LD H,d8 2 8 - - - -	DAA 1 4 Z - 0 C	JR Z,r8 2 12/8 - - - -	ADD HL,HL 1 8 - 0 H C	LD A,(HL+) 1 8 - - - -	DEC HL 1 8 - - - -	INC L 1 4 Z 0 H -	DEC L 1 4 Z 1 H -	LD L,d8 2 8 - - - -	CPL 1 4 - 1 1 -
3x	JR NC,r8 2 12/8 - - - -	LD SP,d16 3 12 - - - -	LD (HL-),A 1 8 - - - -	INC SP 1 8 - - - -	INC (HL) 1 12 Z 0 H -	DEC (HL) 1 12 Z 1 H -	LD (HL),d8 2 12 - - - -	SCF 1 4 - 0 0 1	JR C,r8 2 12/8 - - - -	ADD HL,SP 1 8 - 0 H C	LD A,(HL-) 1 8 - - - -	DEC SP 1 8 - - - -	INC A 1 4 Z 0 H -	DEC A 1 4 Z 1 H -	LD A,d8 2 8 - - - -	CCF 1 4 - 0 0 C
4x	LD B,B 1 4 - - - -	LD B,C 1 4 - - - -	LD B,D 1 4 - - - -	LD B,E 1 4 - - - -	LD B,H 1 4 - - - -	LD B,L 1 4 - - - -	LD B,(HL) 1 8 - - - -	LD B,A 1 4 - - - -	LD C,B 1 4 - - - -	LD C,C 1 4 - - - -	LD C,D 1 4 - - - -	LD C,E 1 4 - - - -	LD C,H 1 4 - - - -	LD C,L 1 4 - - - -	LD C,(HL) 1 8 - - - -	LD C,A 1 4 - - - -
5x	LD D,B 1 4 - - - -	LD D,C 1 4 - - - -	LD D,D 1 4 - - - -	LD D,E 1 4 - - - -	LD D,H 1 4 - - - -	LD D,L 1 4 - - - -	LD D,(HL) 1 8 - - - -	LD D,A 1 4 - - - -	LD E,B 1 4 - - - -	LD E,C 1 4 - - - -	LD E,D 1 4 - - - -	LD E,E 1 4 - - - -	LD E,H 1 4 - - - -	LD E,L 1 4 - - - -	LD E,(HL) 1 8 - - - -	LD E,A 1 4 - - - -
6x	LD H,B 1 4 - - - -	LD H,C 1 4 - - - -	LD H,D 1 4 - - - -	LD H,E 1 4 - - - -	LD H,H 1 4 - - - -	LD H,L 1 4 - - - -	LD H,(HL) 1 8 - - - -	LD H,A 1 4 - - - -	LD L,B 1 4 - - - -	LD L,C 1 4 - - - -	LD L,D 1 4 - - - -	LD L,E 1 4 - - - -	LD L,H 1 4 - - - -	LD L,L 1 4 - - - -	LD L,(HL) 1 8 - - - -	LD L,A 1 4 - - - -
7x	LD (HL),B 1 8 - - - -	LD (HL),C 1 8 - - - -	LD (HL),D 1 8 - - - -	LD (HL),E 1 8 - - - -	LD (HL),H 1 8 - - - -	LD (HL),L 1 8 - - - -	HALT 1 4 - - - -	LD (HL),A 1 8 - - - -	LD A,B 1 4 - - - -	LD A,C 1 4 - - - -	LD A,D 1 4 - - - -	LD A,E 1 4 - - - -	LD A,H 1 4 - - - -	LD A,L 1 4 - - - -	LD A,(HL) 1 8 - - - -	LD A,A 1 4 - - - -
8x	ADD A,B 1 4 Z 0 H C	ADD A,C 1 4 Z 0 H C	ADD A,D 1 4 Z 0 H C	ADD A,E 1 4 Z 0 H C	ADD A,H 1 4 Z 0 H C	ADD A,L 1 4 Z 0 H C	ADD A,(HL) 1 8 Z 0 H C	ADD A,A 1 4 Z 0 H C	ADC A,B 1 4 Z 0 H C	ADC A,C 1 4 Z 0 H C	ADC A,D 1 4 Z 0 H C	ADC A,E 1 4 Z 0 H C	ADC A,H 1 4 Z 0 H C	ADC A,L 1 4 Z 0 H C	ADC A,(HL) 1 8 Z 0 H C	ADC A,A 1 4 Z 0 H C
9x	SUB B 1 4 Z 1 H C	SUB C 1 4 Z 1 H C	SUB D 1 4 Z 1 H C	SUB E 1 4 Z 1 H C	SUB H 1 4 Z 1 H C	SUB L 1 4 Z 1 H C	SUB (HL) 1 8 Z 1 H C	SUB A 1 4 Z 1 H C	SBC A,B 1 4 Z 1 H C	SBC A,C 1 4 Z 1 H C	SBC A,D 1 4 Z 1 H C	SBC A,E 1 4 Z 1 H C	SBC A,H 1 4 Z 1 H C	SBC A,L 1 4 Z 1 H C	SBC A,(HL) 1 8 Z 1 H C	SBC A,A 1 4 Z 1 H C
Ax	AND B 1 4 Z 0 1 0	AND C 1 4 Z 0 1 0	AND D 1 4 Z 0 1 0	AND E 1 4 Z 0 1 0	AND H 1 4 Z 0 1 0	AND L 1 4 Z 0 1 0	AND (HL) 1 8 Z 0 1 0	AND A 1 4 Z 0 1 0	XOR B 1 4 Z 0 0 0	XOR C 1 4 Z 0 0 0	XOR D 1 4 Z 0 0 0	XOR E 1 4 Z 0 0 0	XOR H 1 4 Z 0 0 0	XOR L 1 4 Z 0 0 0	XOR (HL) 1 8 Z 0 0 0	XOR A 1 4 Z 0 0 0
Bx	OR B 1 4 Z 0 0 0	OR C 1 4 Z 0 0 0	OR D 1 4 Z 0 0 0	OR E 1 4 Z 0 0 0	OR H 1 4 Z 0 0 0	OR L 1 4 Z 0 0 0	OR (HL) 1 8 Z 0 0 0	OR A 1 4 Z 0 0 0	CP B 1 4 Z 1 H C	CP C 1 4 Z 1 H C	CP D 1 4 Z 1 H C	CP E 1 4 Z 1 H C	CP H 1 4 Z 1 H C	CP L 1 4 Z 1 H C	CP (HL) 1 8 Z 1 H C	CP A 1 4 Z 1 H C
Cx	RET NZ 1 20/8 - - - -	POP BC 1 12 - - - -	JP NZ,a16 3 16/12 - - - -	JP a16 3 16 - - - -	CALL NZ,a16 3 24/12 - - - -	PUSH BC 1 16 - - - -	ADD A,d8 2 8 Z 0 H C	RST 00H 1 16 - - - -	RET Z 1 20/8 - - - -	RET 1 16 - - - -	JP Z,a16 3 16/12 - - - -	PREFIX CB 1 4 - - - -	CALL Z,a16 3 24/12 - - - -	CALL a16 3 24 - - - -	ADC A,d8 2 8 Z 0 H C	RST 08H 1 16 - - - -
Dx	RET NC 1 20/8 - - - -	POP DE 1 12 - - - -	JP NC,a16 3 16/12 - - - -		CALL NC,a16 3 24/12 - - - -	PUSH DE 1 16 - - - -	SUB d8 2 8 Z 1 H C	RST 10H 1 16 - - - -	RET C 1 20/8 - - - -	RETI 1 16 - - - -	JP C,a16 3 16/12 - - - -		CALL C,a16 3 24/12 - - - -		SBC A,d8 2 8 Z 1 H C	RST 18H 1 16 - - - -
Ex	LDH (a8),A 2 12 - - - -	POP HL 1 12 - - - -	LD (C),A 2 8 - - - -			PUSH HL 1 16 - - - -	AND d8 2 8 Z 0 1 0	RST 20H 1 16 - - - -	ADD SP,r8 2 16 0 0 H C	JP (HL) 1 4 - - - -	LD (a16),A 3 16 - - - -				XOR d8 2 8 Z 0 0 0	RST 28H 1 16 - - - -
Fx	LDH A,(a8) 2 12 - - - -	POP AF 1 12 Z N H C	LD A,(C) 2 8 - - - -	DI 1 4 - - - -		PUSH AF 1 16 - - - -	OR d8 2 8 Z 0 0 0	RST 30H 1 16 - - - -	LD HL,SP+r8 2 12 0 0 H C	LD SP,HL 1 8 - - - -	LD A,(a16) 3 16 - - - -	EI 1 4 - - - -			CP d8 2 8 Z 1 H C	RST 38H 1 16 - - - -

LD A,(a16)

3 16

- - - -

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (BC),A 1 8 - - - -	INC BC 1 8 - - - -	INC B 1 4 Z 0 H -	DEC B 1 4 Z 1 H -	LD B,d8 2 8 - - - -	RLCA 1 4 0 0 0 C	LD (a16),SP 3 20 - - - -	ADD HL,BC 1 8 - 0 H C	LD A,(BC) 1 8 - - - -	DEC BC 1 8 - - - -	INC C 1 4 Z 0 H -	DEC C 1 4 Z 1 H -	LD C,d8 2 8 - - - -	RRCA 1 4 0 0 0 C
1x	STOP 0 2 4 - - - -	LD DE,d16 3 12 - - - -	LD (DE),A 1 8 - - - -	INC DE 1 8 - - - -	INC D 1 4 Z 0 H -	DEC D 1 4 Z 1 H -	LD D,d8 2 8 - - - -	RLA 1 4 0 0 0 C	JR r8 2 12 - - - -	ADD HL,DE 1 8 - 0 H C	LD A,(DE) 1 8 - - - -	DEC DE 1 8 - - - -	INC E 1 4 Z 0 H -	DEC E 1 4 Z 1 H -	LD E,d8 2 8 - - - -	RRA 1 4 0 0 0 C
2x	JR NZ,r8 2 12/8 - - - -	LD HL,d16 3 12 - - - -	LD (HL+),A 1 8 - - - -	INC HL 1 8 - - - -	INC H 1 4 Z 0 H -	DEC H 1 4 Z 1 H -	LD H,d8 2 8 - - - -	DAA 1 4 Z - 0 C	JR Z,r8 2 12/8 - - - -	ADD HL,HL 1 8 - 0 H C	LD A,(HL+) 1 8 - - - -	DEC HL 1 8 - - - -	INC L 1 4 Z 0 H -	DEC L 1 4 Z 1 H -	LD L,d8 2 8 - - - -	CPL 1 4 - 1 1 -
3x	JR NC,r8 2 12/8 - - - -	LD SP,d16 3 12 - - - -	LD (HL-), 1 8 - - - -											DEC A 1 4 Z 1 H -	LD A,d8 2 8 - - - -	CCF 1 4 - 0 0 C
4x	LD B,B 1 4 - - - -	LD B,C 1 4 - - - -	LD B,D 1 4 - - - -											LD C,L 1 4 - - - -	LD C,(HL) 1 8 - - - -	LD C,A 1 4 - - - -
5x	LD D,B 1 4 - - - -	LD D,C 1 4 - - - -	LD D,D 1 4 - - - -											LD E,L 1 4 - - - -	LD E,(HL) 1 8 - - - -	LD E,A 1 4 - - - -
6x	LD H,B 1 4 - - - -	LD H,C 1 4 - - - -	LD H,D 1 4 - - - -											LD L,L 1 4 - - - -	LD L,(HL) 1 8 - - - -	LD L,A 1 4 - - - -
7x	LD (HL),B 1 8 - - - -	LD (HL),C 1 8 - - - -	LD (HL), 1 8 - - - -											LD A,L 1 4 - - - -	LD A,(HL) 1 8 - - - -	LD A,A 1 4 - - - -
8x	ADD A,B 1 4 Z 0 H C	ADD A,C 1 4 Z 0 H C	ADD A,D 1 4 Z 0 H C											ADC A,L 1 4 Z 0 H C	ADC A,(HL) 1 8 Z 0 H C	ADC A,A 1 4 Z 0 H C
9x	SUB B 1 4 Z 1 H C	SUB C 1 4 Z 1 H C	SUB D 1 4 Z 1 H C											SBC A,L 1 4 Z 1 H C	SBC A,(HL) 1 8 Z 1 H C	SBC A,A 1 4 Z 1 H C
Ax	AND B 1 4 Z 0 1 0	AND C 1 4 Z 0 1 0	AND D 1 4 Z 0 1 0											XOR L 1 4 Z 0 0 0	XOR (HL) 1 8 Z 0 0 0	XOR A 1 4 Z 0 0 0
Bx	OR B 1 4 Z 0 0 0	OR C 1 4 Z 0 0 0	OR D 1 4 Z 0 0 0											CP L 1 4 Z 1 H C	CP (HL) 1 8 Z 1 H C	CP A 1 4 Z 1 H C
Cx	RET NZ 1 20/8 - - - -	POP BC 1 12 - - - -	JP NZ,a1 3 16/1 - - - -											CALL a16 3 24 - - - -	ADC A,d8 2 8 Z 0 H C	RST 08H 1 16 - - - -
Dx	RET NC 1 20/8 - - - -	POP DE 1 12 - - - -	JP NC,a16 3 16/12 - - - -		CALL NC,a16 3 24/12 - - - -	PUSH DE 1 16 - - - -	SUB d8 2 8 Z 1 H C	RST 10H 1 16 - - - -	RET C 1 20/8 - - - -	RETI 1 16 - - - -	JP C,a16 3 16/12 - - - -		CALL C,a16 3 24/12 - - - -		SBC A,d8 2 8 Z 1 H C	RST 18H 1 16 - - - -
Ex	LDH (a8),A 2 12 - - - -	POP HL 1 12 - - - -	LD (C),A 2 8 - - - -			PUSH HL 1 16 - - - -	AND d8 2 8 Z 0 1 0	RST 20H 1 16 - - - -	ADD SP,r8 2 16 0 0 H C	JP (HL) 1 4 - - - -	LD (a16),A 3 16 - - - -				XOR d8 2 8 Z 0 0 0	RST 28H 1 16 - - - -
Fx	LDH A,(a8) 2 12 - - - -	POP AF 1 12 Z N H C	LD A,(C) 2 8 - - - -	DI 1 4 - - - -		PUSH AF 1 16 - - - -	OR d8 2 8 Z 0 0 0	RST 30H 1 16 - - - -	LD HL,SP+r8 2 12 0 0 H C	LD SP,HL 1 8 - - - -	LD A,(a16) 3 16 - - - -	EI 1 4 - - - -			CP d8 2 8 Z 1 H C	RST 38H 1 16 - - - -

LD A,(a16)

3	16	4
bytes	clocks @ 4 MHz	clocks @ 1 MHz

_ _ _ _

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - -	LD BC,d16 3 12 - - - -	LD (BC),A 1 8 - - - -	INC BC 1 8 - - - -	INC B 1 4 Z 0 H -	DEC B 1 4 Z 1 H -	LD B,d8 2 8 - - - -	RLCA 1 4 0 0 0 C	LD (a16),SP 3 20 - - - -	ADD HL,BC 1 8 - 0 H C	LD A,(BC) 1 8 - - - -	DEC BC 1 8 - - - -	INC C 1 4 Z 0 H -	DEC C 1 4 Z 1 H -	LD C,d8 2 8 - - - -	RRCA 1 4 0 0 0 C
1x	STOP 0 2 4 - - - -	LD DE,d16 3 12 - - - -	LD (DE),A 1 8 - - - -	INC DE 1 8 - - - -	INC D 1 4 Z 0 H -	DEC D 1 4 Z 1 H -	LD D,d8 2 8 - - - -	RLA 1 4 0 0 0 C	JR r8 2 12 - - - -	ADD HL,DE 1 8 - 0 H C	LD A,(DE) 1 8 - - - -	DEC DE 1 8 - - - -	INC E 1 4 Z 0 H -	DEC E 1 4 Z 1 H -	LD E,d8 2 8 - - - -	RRA 1 4 0 0 0 C
2x	JR NZ,r8 2 12/8 - - - -	LD HL,d16 3 12 - - - -	LD (HL+),A 1 8 - - - -	INC HL 1 8 - - - -	INC H 1 4 Z 0 H -	DEC H 1 4 Z 1 H -	LD H,d8 2 8 - - - -	DAA 1 4 Z - 0 C	JR Z,r8 2 12/8 - - - -	ADD HL,HL 1 8 - 0 H C	LD A,(HL+) 1 8 - - - -	DEC HL 1 8 - - - -	INC L 1 4 Z 0 H -	DEC L 1 4 Z 1 H -	LD L,d8 2 8 - - - -	CPL 1 4 - 1 1 -
3x	JR NC,r8 2 12/8 - - - -	LD SP,d16 3 12 - - - -	LD (HL-) 1 8 - - - -											DEC A 1 4 Z 1 H -	LD A,d8 2 8 - - - -	CCF 1 4 - 0 0 C
4x	LD B,B 1 4 - - - -	LD B,C 1 4 - - - -	LD B,D 1 4 - - - -											LD C,L 1 4 - - - -	LD C,(HL) 1 8 - - - -	LD C,A 1 4 - - - -
5x	LD D,B 1 4 - - - -	LD D,C 1 4 - - - -	LD D,D 1 4 - - - -											LD E,L 1 4 - - - -	LD E,(HL) 1 8 - - - -	LD E,A 1 4 - - - -
6x	LD H,B 1 4 - - - -	LD H,C 1 4 - - - -	LD H,D 1 4 - - - -											LD L,L 1 4 - - - -	LD L,(HL) 1 8 - - - -	LD L,A 1 4 - - - -
7x	LD (HL),B 1 8 - - - -	LD (HL),C 1 8 - - - -	LD (HL), 1 8 - - - -											LD A,L 1 4 - - - -	LD A,(HL) 1 8 - - - -	LD A,A 1 4 - - - -
8x	ADD A,B 1 4 Z 0 H C	ADD A,C 1 4 Z 0 H C	ADD A,D 1 4 Z 0 H C											ADC A,L 1 4 Z 0 H C	ADC A,(HL) 1 8 Z 0 H C	ADC A,A 1 4 Z 0 H C
9x	SUB B 1 4 Z 1 H C	SUB C 1 4 Z 1 H C	SUB D 1 4 Z 1 H C											SBC A,L 1 4 Z 1 H C	SBC A,(HL) 1 8 Z 1 H C	SBC A,A 1 4 Z 1 H C
Ax	AND B 1 4 Z 0 1 0	AND C 1 4 Z 0 1 0	AND D 1 4 Z 0 1 0											XOR L 1 4 Z 0 0 0	XOR (HL) 1 8 Z 0 0 0	XOR A 1 4 Z 0 0 0
Bx	OR B 1 4 Z 0 0 0	OR C 1 4 Z 0 0 0	OR D 1 4 Z 0 0 0											CP L 1 4 Z 1 H C	CP (HL) 1 8 Z 1 H C	CP A 1 4 Z 1 H C
Cx	RET NZ 1 20/8 - - - -	POP BC 1 12 - - - -	JP NZ,a1 3 16/1 - - - -											CALL a16 3 24 - - - -	ADC A,d8 2 8 Z 0 H C	RST 08H 1 16 - - - -
Dx	RET NC 1 20/8 - - - -	POP DE 1 12 - - - -	JP NC,a16 3 16/12 - - - -		CALL NC,a16 3 24/12 - - - -	PUSH DE 1 16 - - - -	SUB d8 2 8 Z 1 H C	RST 10H 1 16 - - - -	RET C 1 20/8 - - - -	RETI 1 16 - - - -	JP C,a16 3 16/12 - - - -		CALL C,a16 3 24/12 - - - -		SBC A,d8 2 8 Z 1 H C	RST 18H 1 16 - - - -
Ex	LDH (a8),A 2 12 - - - -	POP HL 1 12 - - - -	LD (C),A 2 8 - - - -			PUSH HL 1 16 - - - -	AND d8 2 8 Z 0 1 0	RST 20H 1 16 - - - -	ADD SP,r8 2 16 0 0 H C	JP (HL) 1 4 - - - -	LD (a16),A 3 16 - - - -				XOR d8 2 8 Z 0 0 0	RST 28H 1 16 - - - -
Fx	LDH A,(a8) 2 12 - - - -	POP AF 1 12 Z N H C	LD A,(C) 2 8 - - - -	DI 1 4 - - - -		PUSH AF 1 16 - - - -	OR d8 2 8 Z 0 0 0	RST 30H 1 16 - - - -	LD HL,SP+r8 2 12 0 0 H C	LD SP,HL 1 8 - - - -	LD A,(a16) 3 16 - - - -	EI 1 4 - - - -			CP d8 2 8 Z 1 H C	RST 38H 1 16 - - - -

System Clocks

System Clocks

CPU	4 MHz

System Clocks

CPU	4 MHz
RAM	1 MHz

System Clocks

CPU	4 MHz
RAM	1 MHz
PPU	4 MHz

System Clocks

CPU	4 MHz
RAM	1 MHz
PPU	4 MHz
VRAM	2 MHz

System Clocks

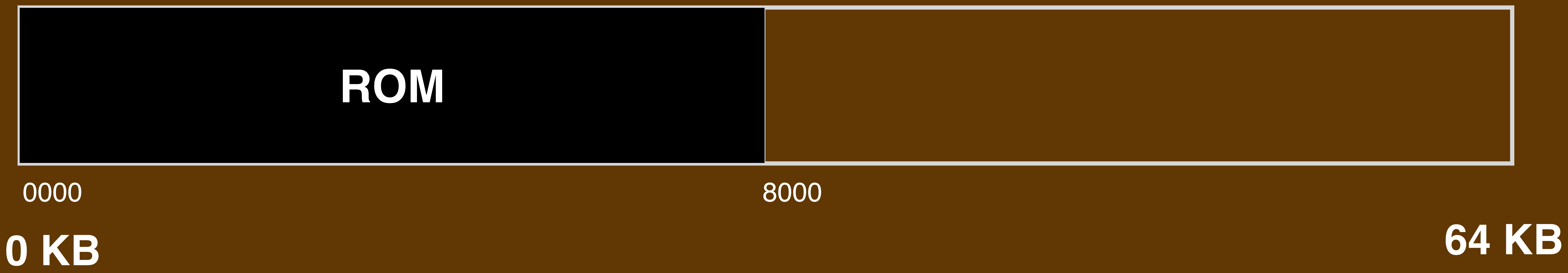
CPU	4,194,304 Hz
RAM	1,048,576 Hz
PPU	4,194,304 Hz
VRAM	2,097,152 Hz

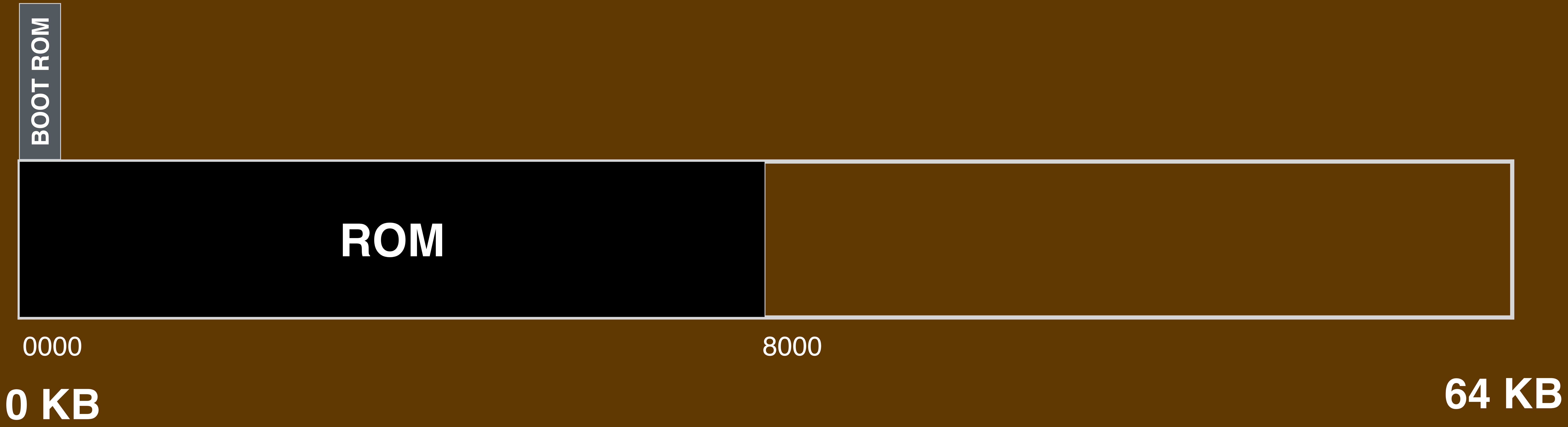
System Clocks

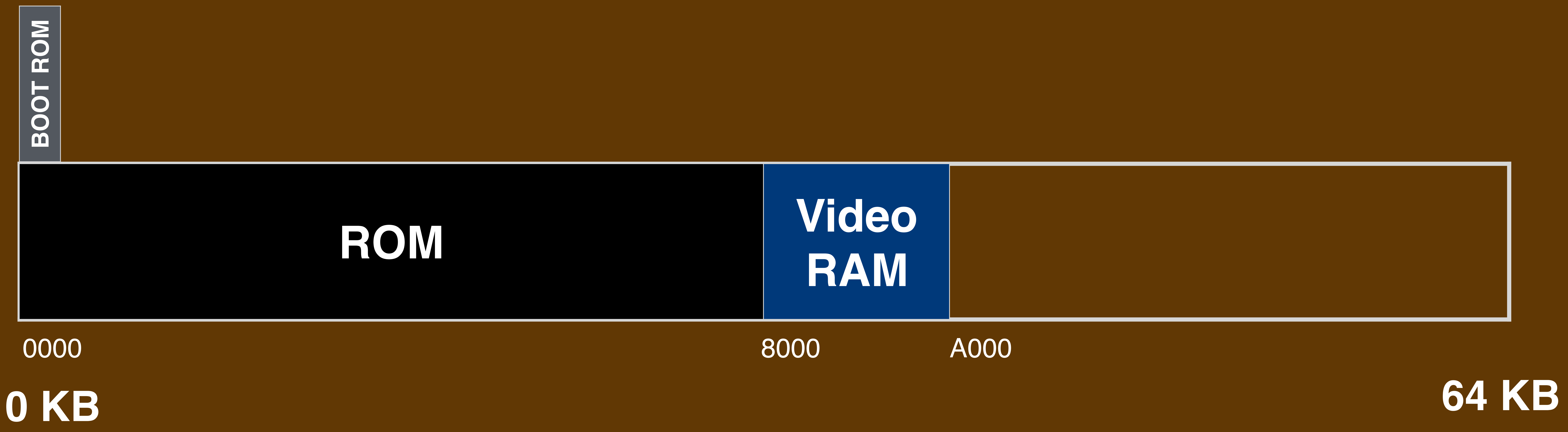
CPU	4,194,304 Hz
RAM	1,048,576 Hz
PPU	4,194,304 Hz
VRAM	2,097,152 Hz

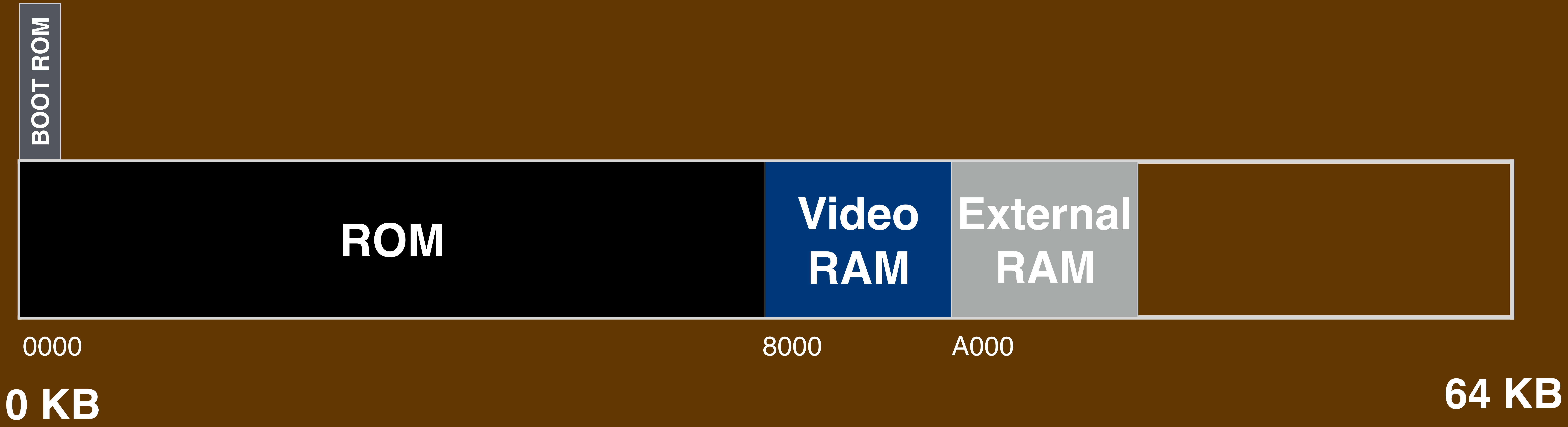
"machine cycle" @ 1 MHz

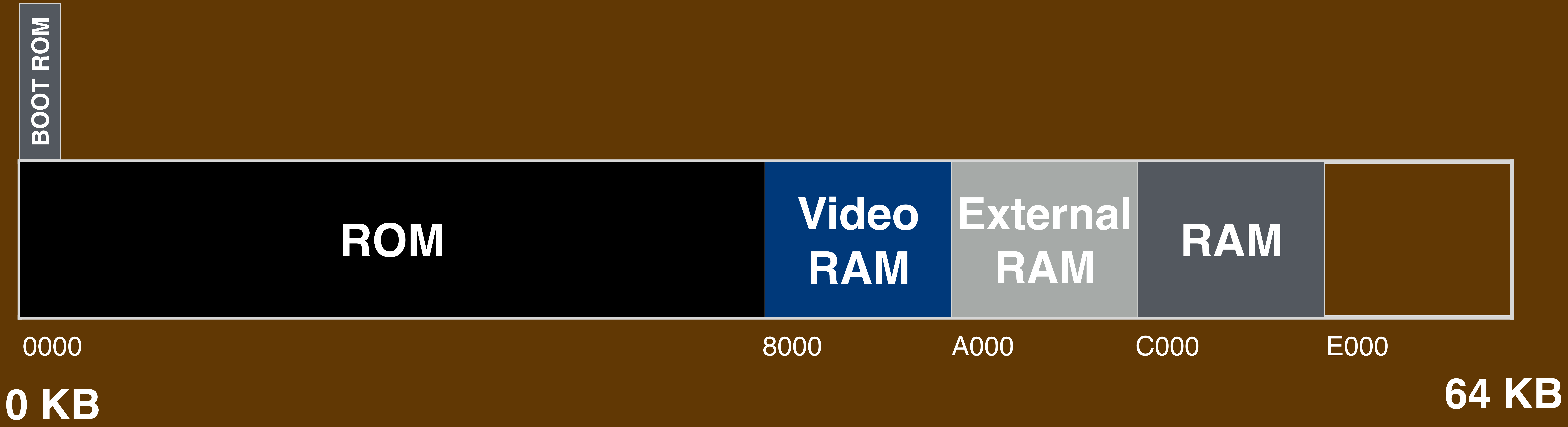
0 KB **64 KB**

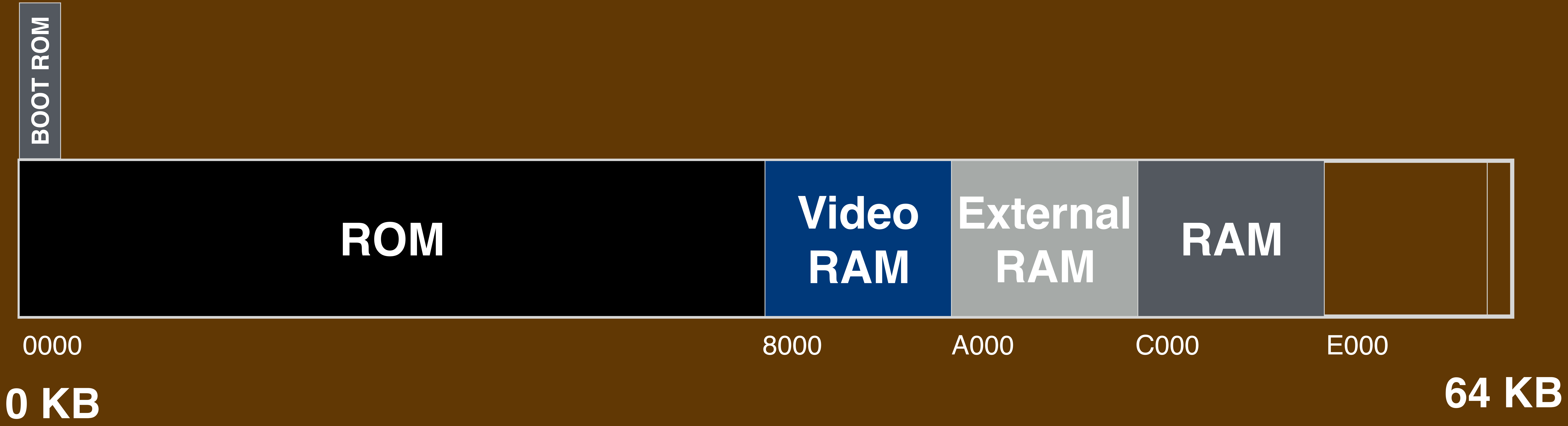

ROM
0000
8000
0 KB
64 KB


BOOT ROM
ROM
0000
8000
0 KB
64 KB

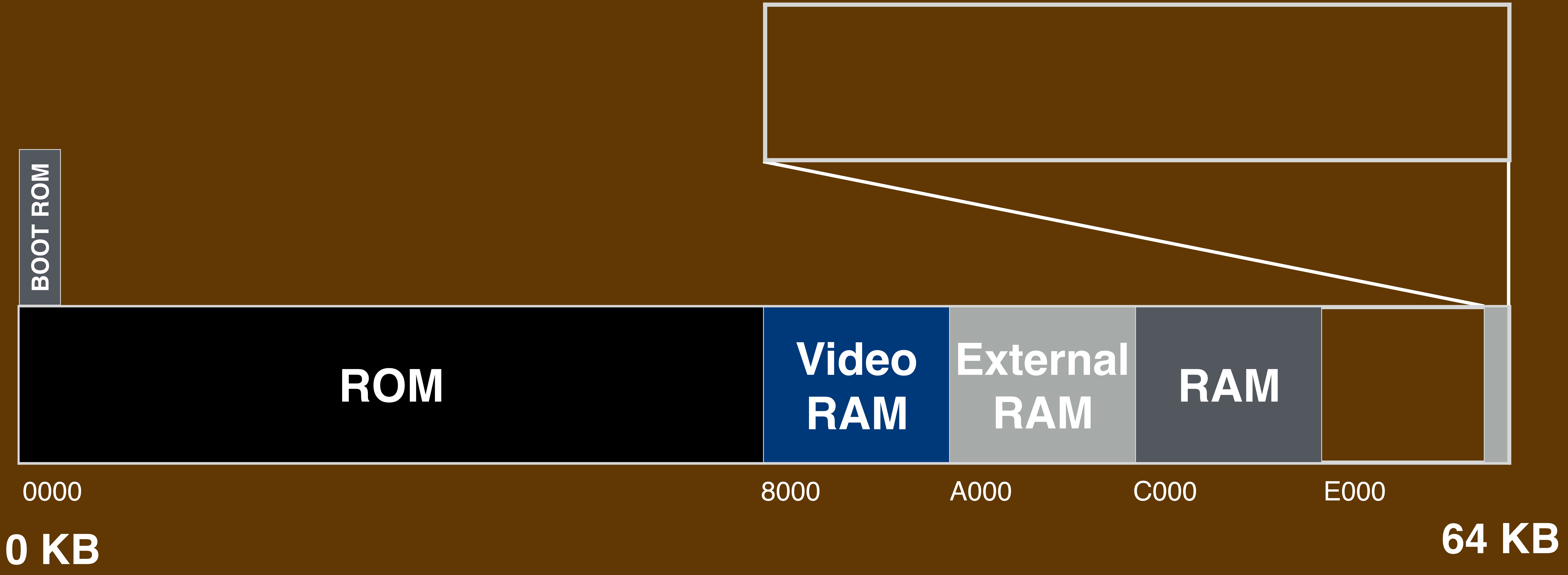

BOOT ROM
ROM
Video RAM
0000
8000
A000
0 KB
64 KB


BOOT ROM
ROM
Video RAM
External RAM
0000
8000
A000
0 KB
64 KB

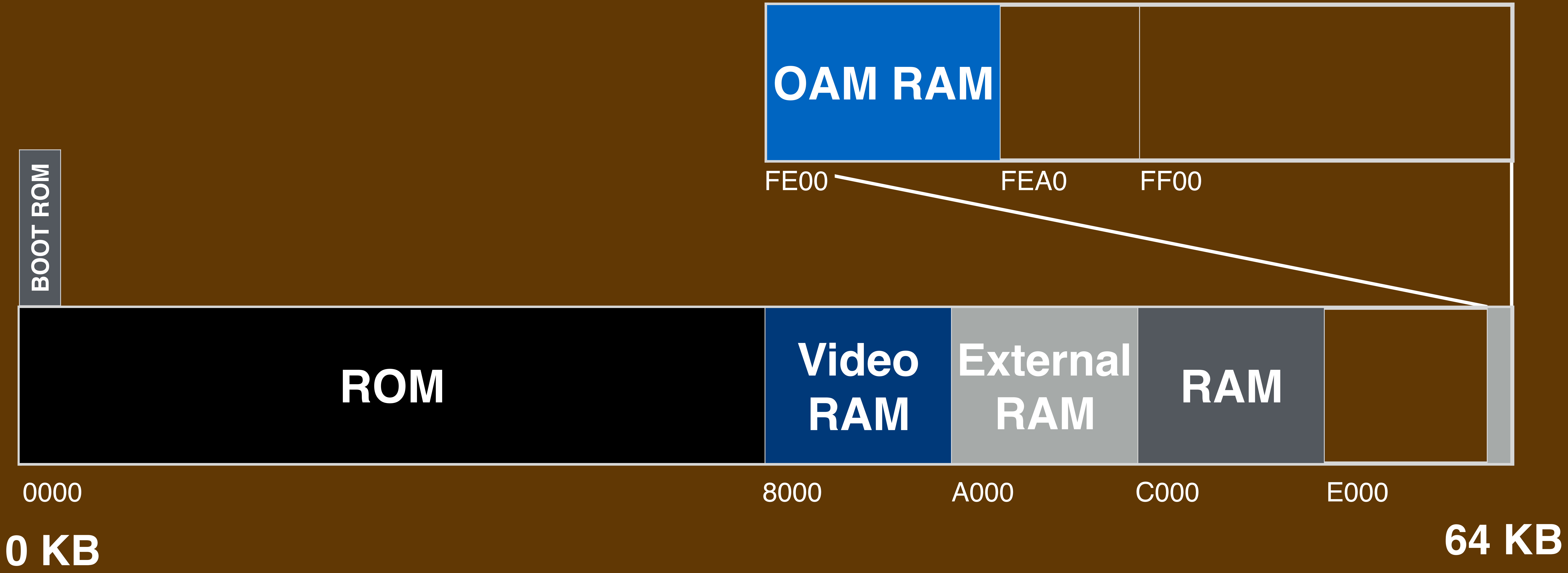

BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB

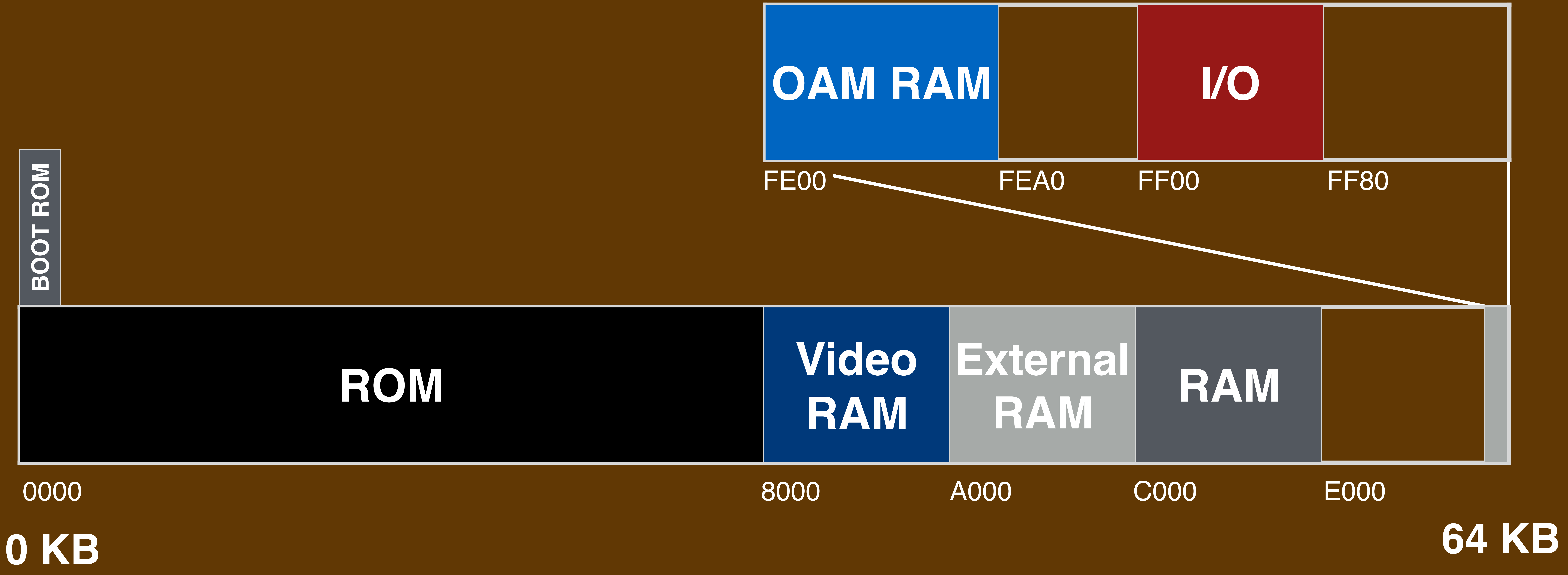

BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB

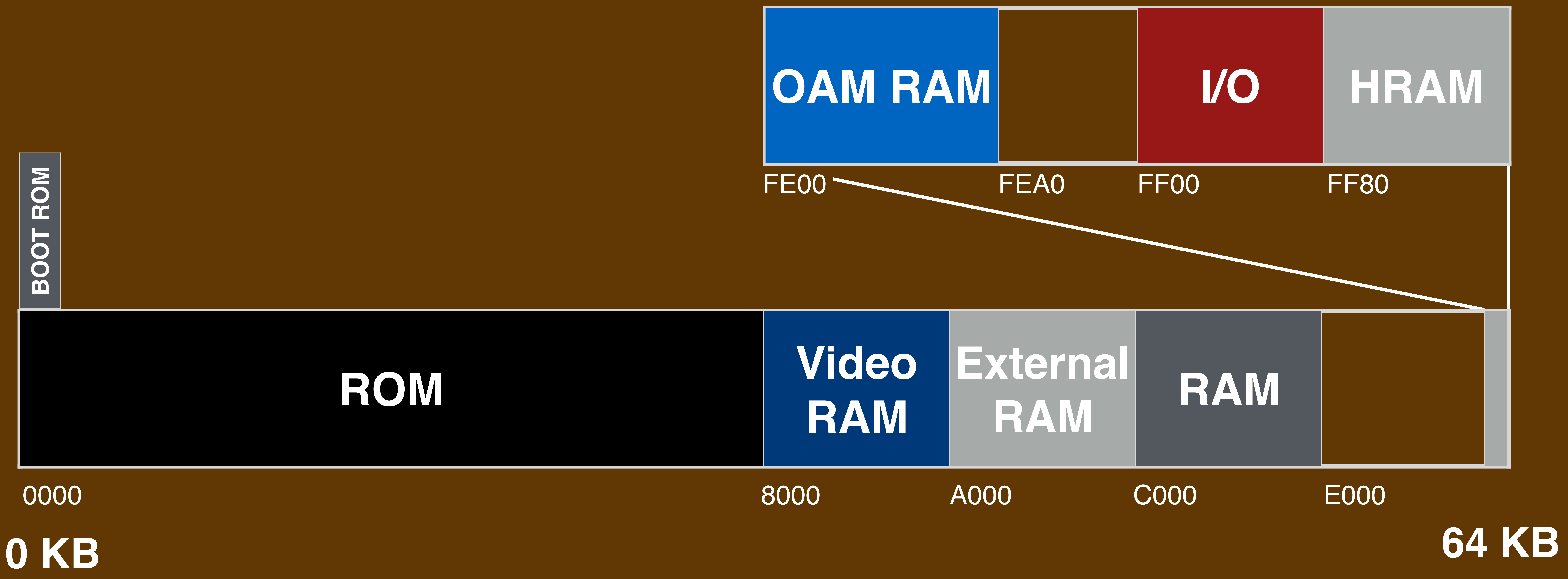

BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB


BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB


OAM RAM
FE00
FEA0
BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB

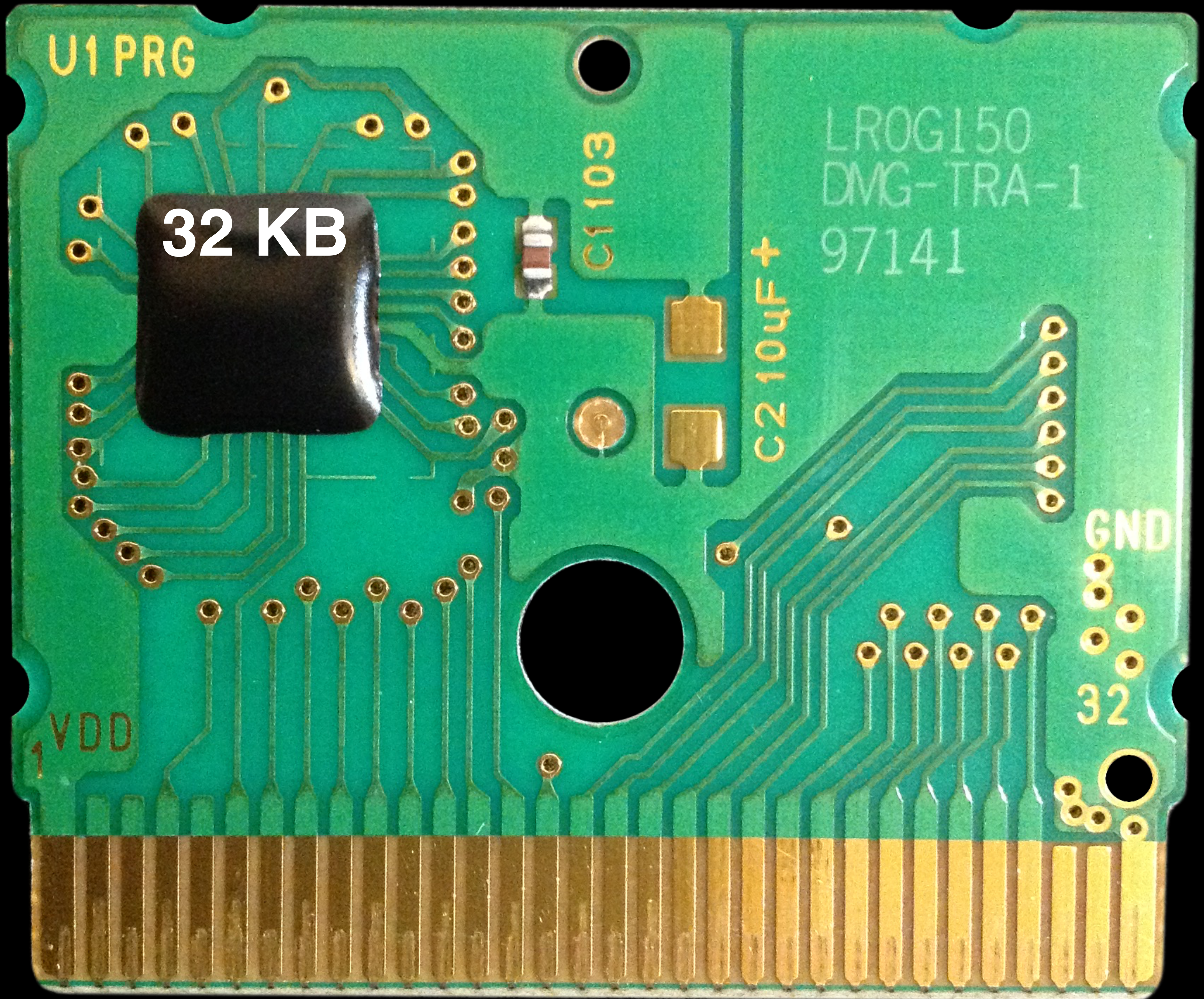

OAM RAM
FE00
FEA0
FF00
BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB


OAM RAM
I/O
FE00
FEA0
FF00
FF80
BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB

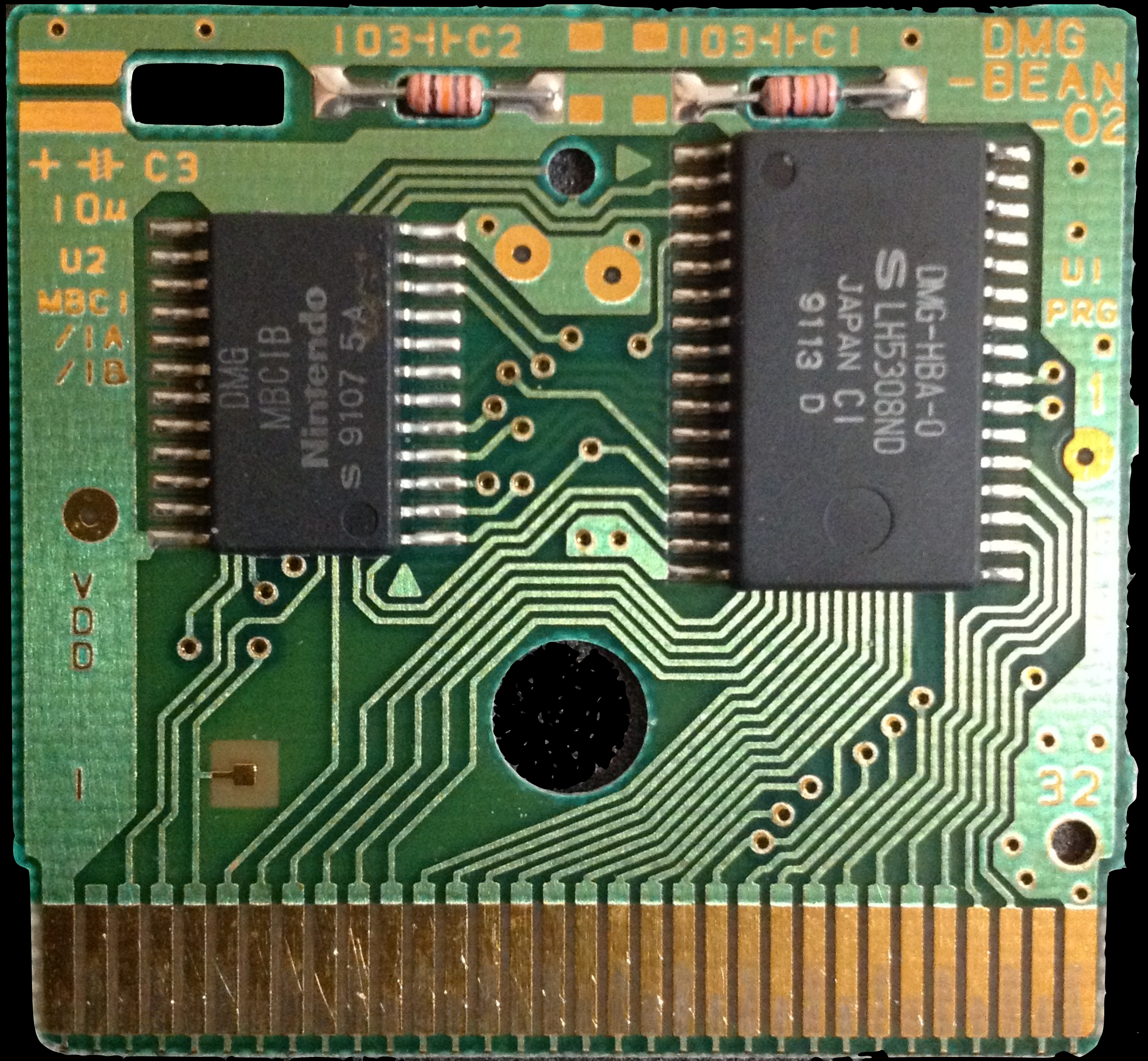

OAM RAM
I/O
HRAM
FE00
FEA0
FF00
FF80
BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB

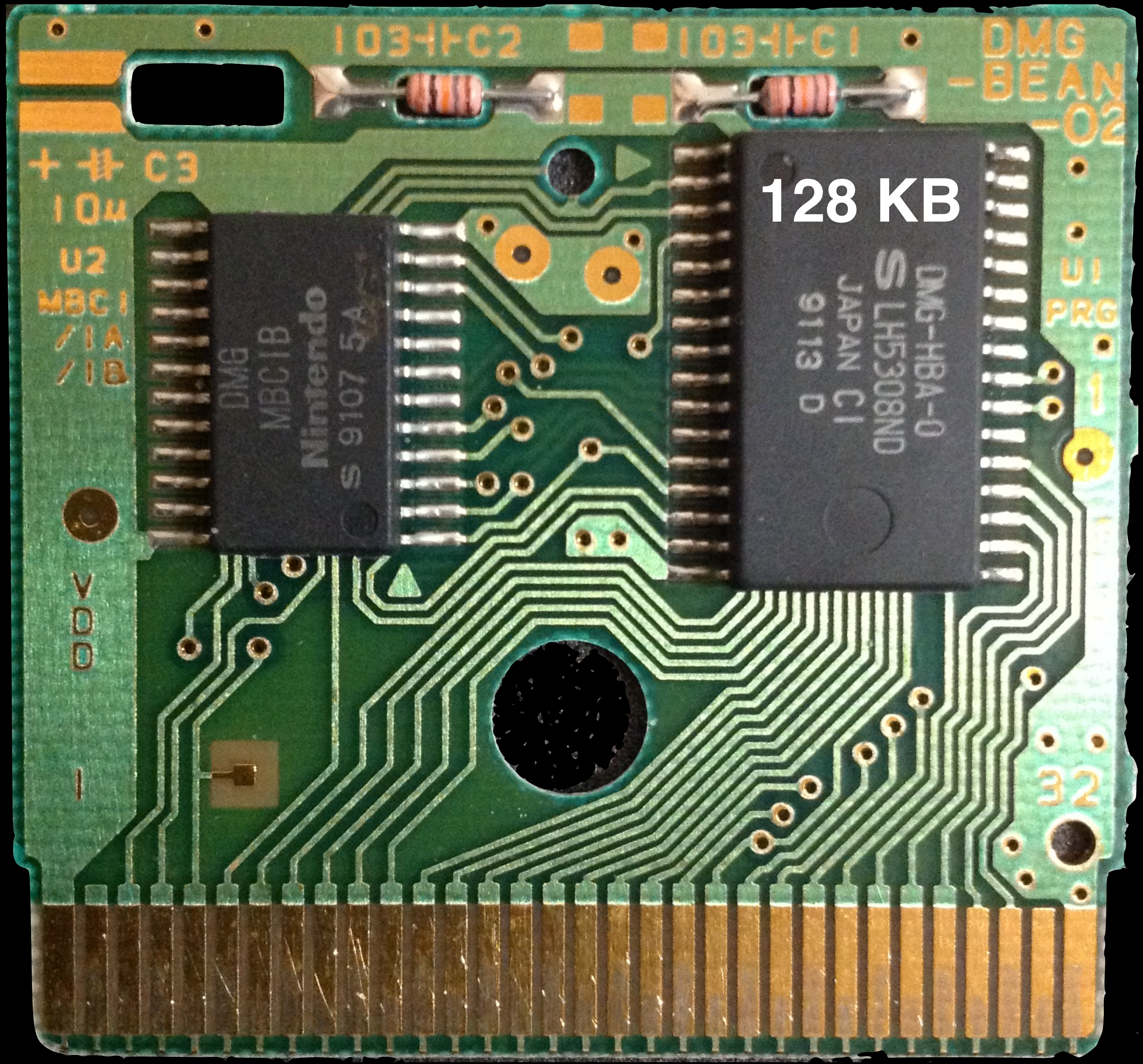
32 KB?

https://gekkio.fi/blog/

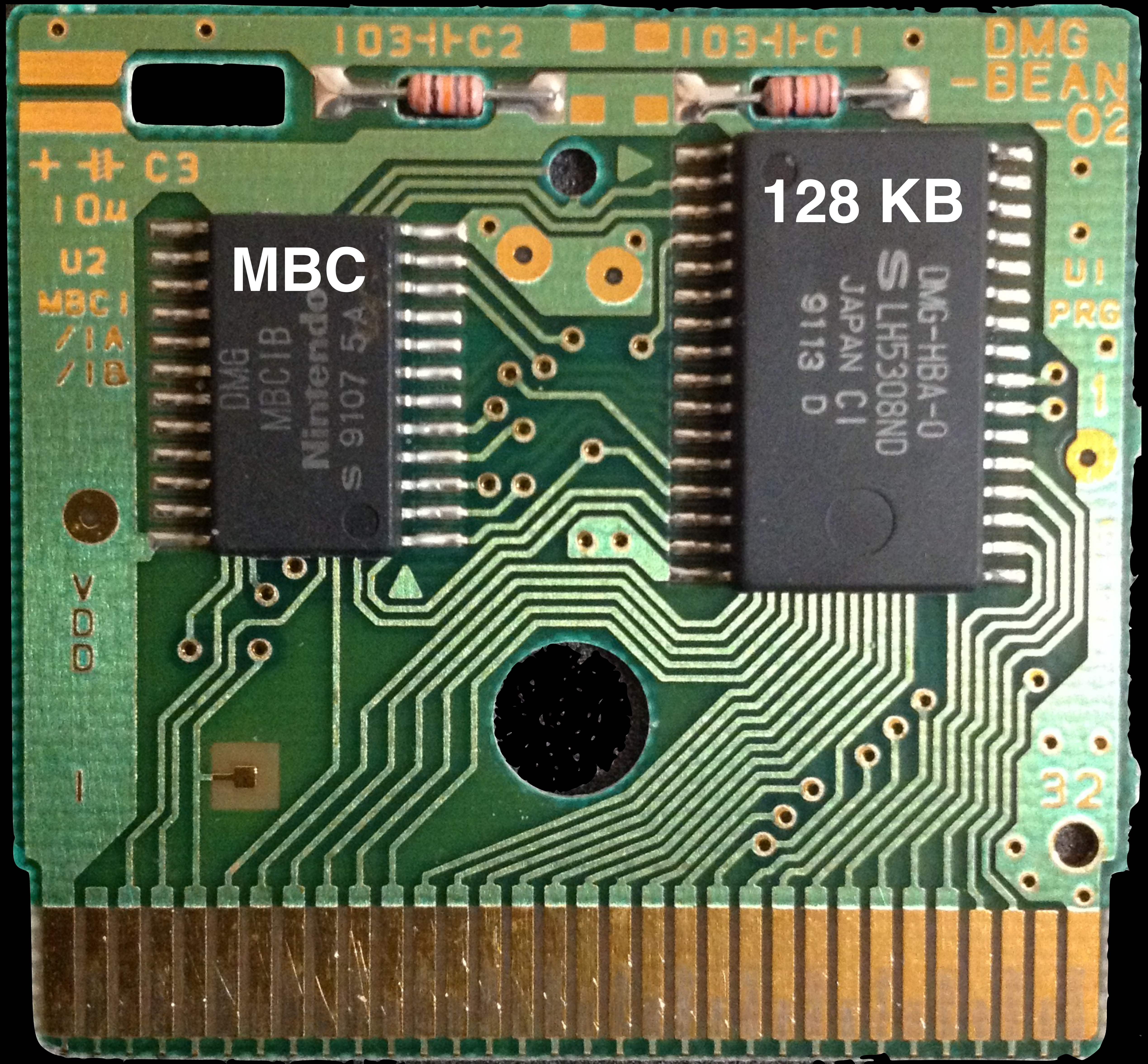


https://gekkio.fi/blog/


C2
C1
DMG
-BEAN
-02
C3
U2
MBC1
/1A
/1B
DMG
MBC1B
Nintendo
S 9107 5A
DMG-HBA-0
S LH5308ND
JAPAN C1
9113 D
U1
PRG
1
32



https://gekkio.fi/blog/


MBC
128 KB
DMG
-BEAN
-02
DMG-HBA-0
LH5308ND
JAPAN C1
9113 D

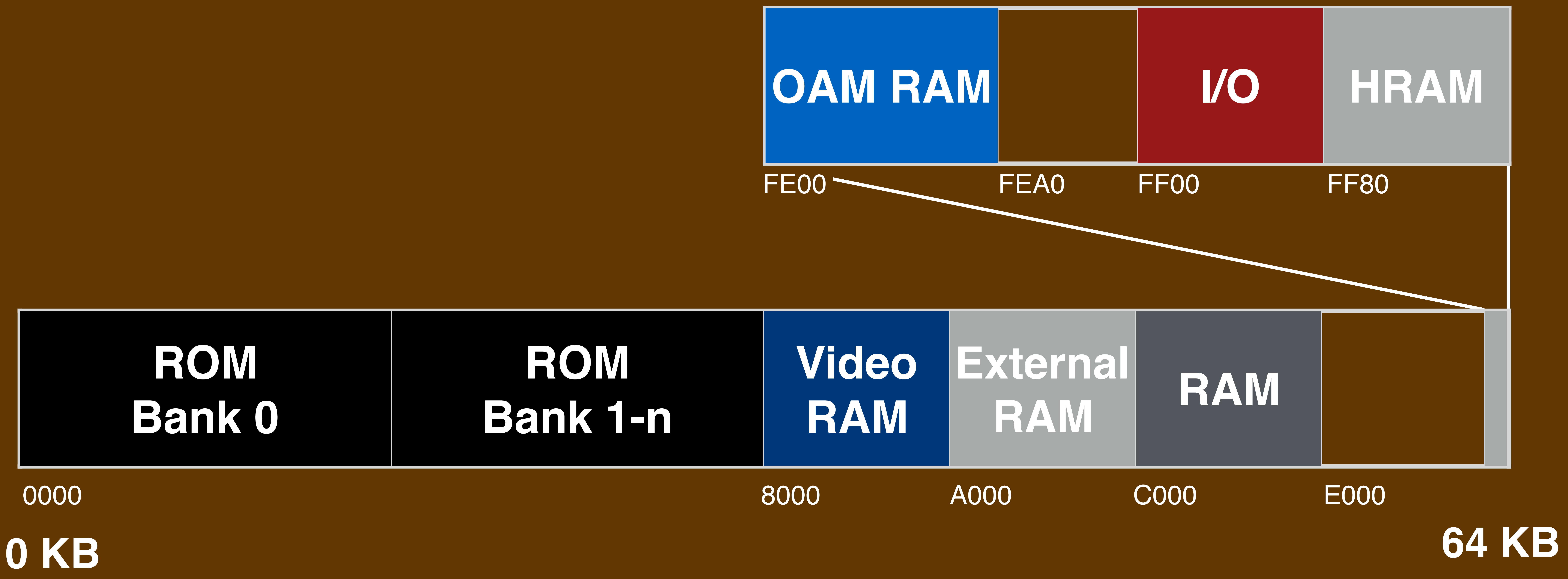

ROM
0000
800

ROM
Bank 0

ROM
Bank 1

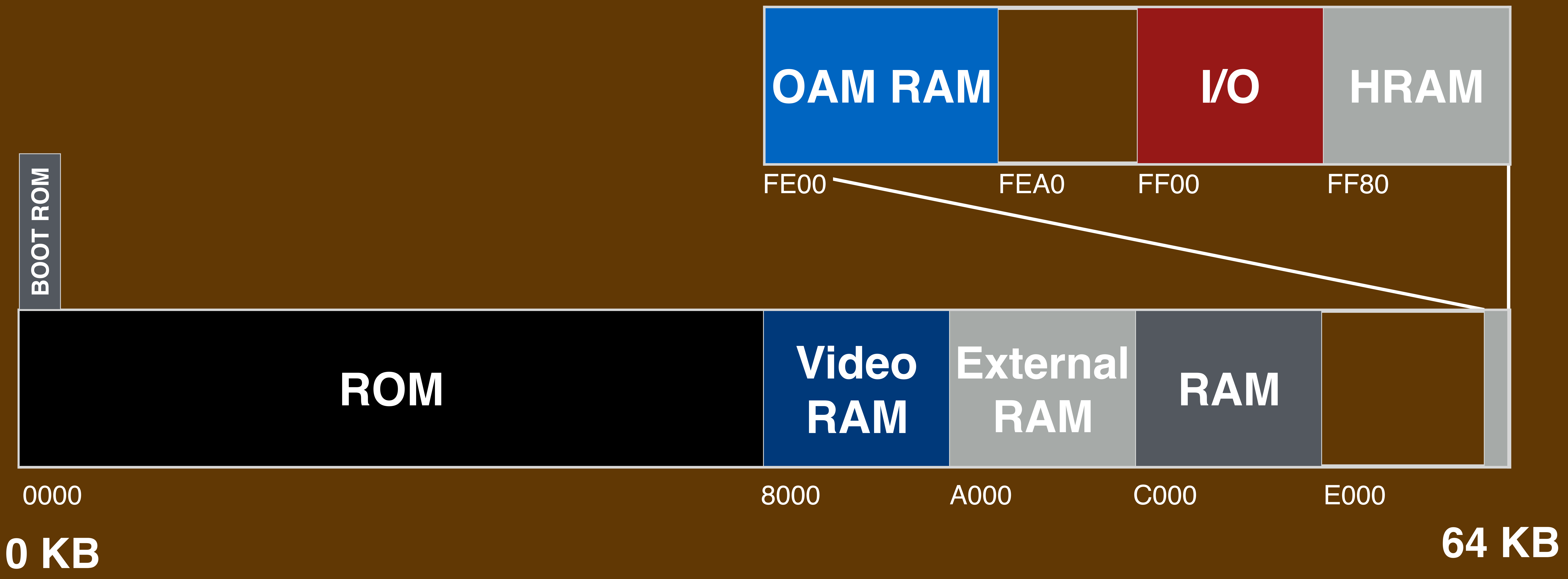
0000 4000 8000

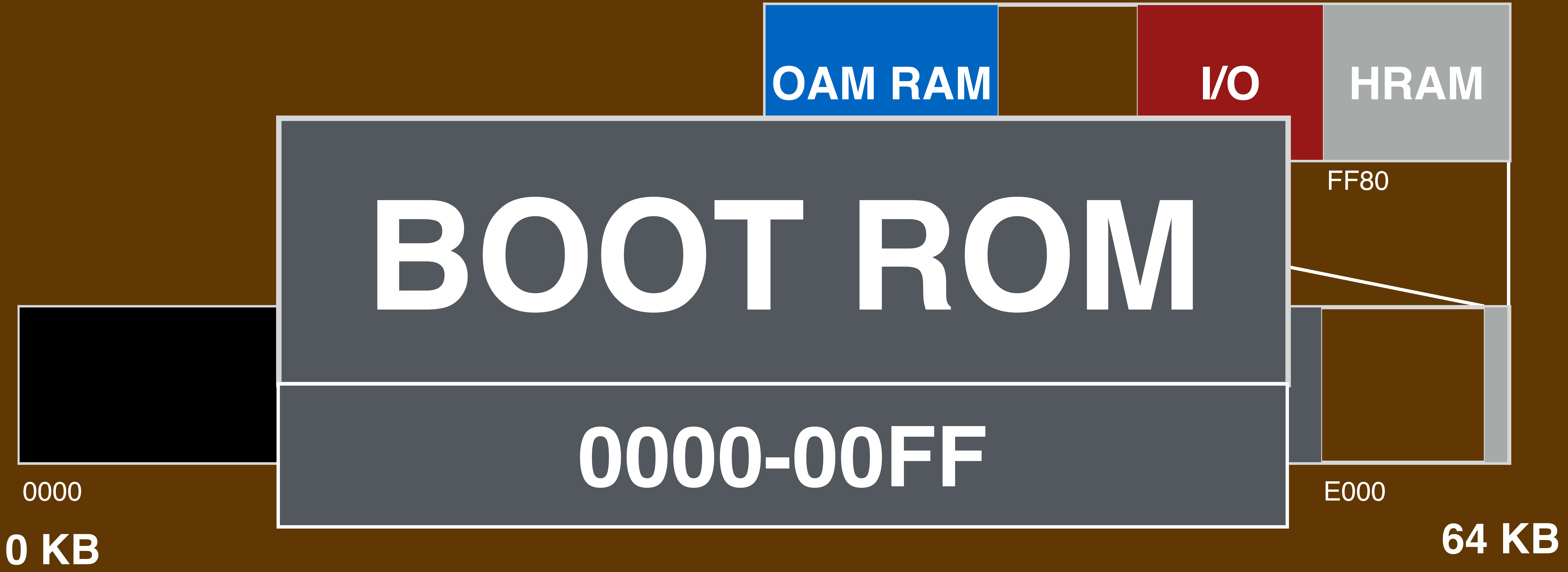

Bank 3
ROM
Bank 2
ROM
Bank 0
ROM
Bank 1
0000
4000


OAM RAM
I/O
HRAM
FE00
FEA0
FF00
FF80
ROM
Bank 0
ROM
Bank 1-n
Video
RAM
External
RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB


Video RAM
External RAM 0
RAM
A000
C000


RAM 2
External
RAM 1
Video
RAM
External
RAM 0
RAM
A000
C000


OAM RAM
I/O
HRAM
FE00
FEA0
FF00
FF80
BOOT ROM
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB


OAM RAM
I/O
HRAM
FF80
BOOT ROM
0000-00FF
0000
E000
0 KB
64 KB

https://www.youtube.com/watch?v=QyvBNhzshKQ

Nintendo®

Nintendo®

Boot ROM

```
	ld sp,$fffe
	xor a
	ld hl,$9fff
addr_0007:
	ld (hl-),a
	bit 7,h
	jr nz, addr_0007
	ld hl,$ff26
	ld c,$11
	ld a,$80
	ld (hl-),a
	ld ($ff00+c),a
	inc c
	ld a,$f3
	ld ($ff00+c),a
	ld (hl-),a
	ld a,$77
	ld (hl),a
	ld a,$fc
	ld ($ff00+$47),a
	ld de,$0104
	ld hl,$8010
addr_0027:
	ld a,(de)
	call $0095
	call $0096
	inc de
	ld a,e
	cp $34
	jr nz, addr_0027
	ld de,$00d8
	ld b,$08
addr_0039:
	ld a,(de)
	inc de
	ld (hl+),a
	inc hl
	dec b
	jr nz, addr_0039
	ld a,$19
	ld ($9910),a
	ld hl,$992f
addr_0048:
	ld c,$0c
addr_004a:
	dec a
	jr z, addr_0055
	ld (hl-),a
	dec c
	jr nz, addr_004a
	ld l,$0f
	jr addr_0048
addr_0055:
	ld h,a
	ld a,$64
	ld d,a
	ld ($ff00+$42),a
	ld a,$91
	ld ($ff00+$40),a
	inc b
addr_0060:
	ld e,$02
addr_0062:
	ld c,$0c
addr_0064:
	ld a,($ff00+$44)
	cp $90
	jr nz, addr_0064
	dec c
	jr nz, addr_0064
	dec e
	jr nz, addr_0062
	ld c,$13
	inc h
	ld a,h
	ld e,$83
	cp $62
	jr z, addr_0080
	ld e,$c1
	cp $64
	jr nz, addr_0086
addr_0080:
	ld a,e
	ld ($ff00+c),a
	inc c
	ld a,$87
	ld ($ff00+c),a
addr_0086:
	ld a,($ff00+$42)
	sub b
	ld ($ff00+$42),a
	dec d
	jr nz, addr_0060
	dec b
	jr nz, addr_00e0
	ld d,$20
	jr addr_0060
	ld c,a
	ld b,$04
addr_0098:
	push bc
	rl c
	rla
	pop bc
	rl c
	rla
	dec b
	jr nz, addr_0098
	ld (hl+),a
	inc hl
	ld (hl+),a
	inc hl
	ret
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
addr_00e0:
	ld hl,$0104
	ld de,$00a8
addr_00e6:
	ld a,(de)
	inc de
	cp (hl)
	jr nz,$fe
	inc hl
	ld a,l
	cp $34
	jr nz, addr_00e6
	ld b,$19
	ld a,b
addr_00f4:
	add (hl)
	inc hl
	dec b
	jr nz, addr_00f4
	add (hl)
	jr nz,$fe
	ld a,$01
	ld ($ff00+$50),a
```

Boot ROM

Init RAM

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec b
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Boot ROM

Init RAM

Init Sound

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec b
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Boot ROM

Init RAM

Init Sound

Set up Logo

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
```

```
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec b
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
```

Decode logo

```
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Boot ROM

Init RAM

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
```

Init Sound

```
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
```

Set up Logo

```
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
```

Scroll logo

```
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
```

Scroll logo

```
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec e
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
```

Decode logo

```
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
```

```
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Boot ROM

Init RAM

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
```

Init Sound

```
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
```

Set up Logo

```
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
```

Scroll logo

```
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
```

Play Sound

```
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
```

Scroll logo

```
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec b
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
```

Decode logo

```
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
```

```
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Boot ROM

Init RAM

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
```

Init Sound

```
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
```

Set up Logo

```
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
```

Scroll logo

```
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
```

Play Sound

```
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
```

Scroll logo

```
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec b
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
```

Decode logo

```
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
```

Logo data

```
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
```

Compare logo

```
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Boot ROM

Init RAM

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
```

Init Sound

```
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
```

Set up Logo

```
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
```

Scroll logo

```
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
```

Play Sound

```
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
```

Scroll logo

```
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec b
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
```

Decode logo

```
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
```

Logo data

```
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
```

Compare logo

```
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
```

Checksum header

```
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Boot ROM

Init RAM

```
        ld sp,$fffe
        xor a
        ld hl,$9fff
addr_0007:
        ld (hl-),a
        bit 7,h
        jr nz, addr_0007
```

Init Sound

```
        ld hl,$ff26
        ld c,$11
        ld a,$80
        ld (hl-),a
        ld ($ff00+c),a
        inc c
        ld a,$f3
        ld ($ff00+c),a
        ld (hl-),a
        ld a,$77
        ld (hl),a
```

Set up Logo

```
        ld a,$fc
        ld ($ff00+$47),a
        ld de,$0104
        ld hl,$8010
addr_0027:
        ld a,(de)
        call $0095
        call $0096
        inc de
        ld a,e
        cp $34
        jr nz, addr_0027
        ld de,$00d8
        ld b,$08
addr_0039:
        ld a,(de)
        inc de
        ld (hl+),a
        inc hl
        dec b
        jr nz, addr_0039
        ld a,$19
        ld ($9910),a
        ld hl,$992f
addr_0048:
        ld c,$0c
addr_004a:
        dec a
        jr z, addr_0055
        ld (hl-),a
        dec c
        jr nz, addr_004a
        ld l,$0f
        jr addr_0048
```

Scroll logo

```
addr_0055:
        ld h,a
        ld a,$64
        ld d,a
        ld ($ff00+$42),a
        ld a,$91
        ld ($ff00+$40),a
        inc b
addr_0060:
        ld e,$02
addr_0062:
        ld c,$0c
addr_0064:
        ld a,($ff00+$44)
        cp $90
        jr nz, addr_0064
        dec c
        jr nz, addr_0064
        dec e
        jr nz, addr_0062
        ld c,$13
        inc h
        ld a,h
        ld e,$83
        cp $62
        jr z, addr_0080
        ld e,$c1
        cp $64
        jr nz, addr_0086
```

Play Sound

```
addr_0080:
        ld a,e
        ld ($ff00+c),a
        inc c
        ld a,$87
        ld ($ff00+c),a
```

Scroll logo

```
addr_0086:
        ld a,($ff00+$42)
        sub b
        ld ($ff00+$42),a
        dec d
        jr nz, addr_0060
        dec b
        jr nz, addr_00e0
        ld d,$20
        jr addr_0060
```

Decode logo

```
        ld c,a
        ld b,$04
addr_0098:
        push bc
        rl c
        rla
        pop bc
        rl c
        rla
        dec b
        jr nz, addr_0098
        ld (hl+),a
        inc hl
        ld (hl+),a
        inc hl
        ret
```

Logo data

```
addr_00a8:
.db $ce,$ed,$66,$66,$cc,$0d,$00,$0b
.db $03,$73,$00,$83,$00,$0c,$00,$0d
.db $00,$08,$11,$1f,$88,$89,$00,$0e
.db $dc,$cc,$6e,$e6,$dd,$dd,$d9,$99
.db $bb,$bb,$67,$63,$6e,$0e,$ec,$cc
.db $dd,$dc,$99,$9f,$bb,$b9,$33,$3e
addr_00d8:
.db $3c,$42,$b9,$a5,$b9,$a5,$42,$3c
```

Compare logo

```
addr_00e0:
        ld hl,$0104
        ld de,$00a8
addr_00e6:
        ld a,(de)
        inc de
        cp (hl)
        jr nz,$fe
        inc hl
        ld a,l
        cp $34
        jr nz, addr_00e6
```

Checksum header

```
        ld b,$19
        ld a,b
addr_00f4:
        add (hl)
        inc hl
        dec b
        jr nz, addr_00f4
        add (hl)
        jr nz,$fe
        ld a,$01
        ld ($ff00+$50),a
```

Turn off ROM

Nintendo®

[redacted]®

Logo Hack

Logo Hack

- **Is That a Demo in Your Pocket?**
- **Gejmbåj**
- **Oh!**
- **20y**

by **Snorpung**

https://vimeo.com/user1849607

Logo Hack

- **Is That a Demo in Your Pocket?**
- **Gejmbåj**
- **Oh!**
- **20y**

by **Snorpung**

https://vimeo.com/user1849607

Boot ROM

```
00f4 86          add (hl)
00f5 23          inc hl
00f6 05          dec b
00f7 20 fb       jr nz, $00f4
00f9 86          add (hl)
00fa 20 fe       jr nz, $fe
00fc 3e 01       ld a, $01
00fe e0 50       ld ($ff00+$50), a
```

Game ROM

```
0100 00          nop
0101 c3 50 01    jp $0150
0104             ...
  .              ...
  .              ...
  .              ...
  .              ...
```

Boot ROM

```
00f4 86        add (hl)
00f5 23        inc hl
00f6 05        dec b
00f7 20 fb     jr nz, $00f4
00f9 86        add (hl)
00fa 20 fe     jr nz, $fe
00fc 3e 01     ld a, $01
00fe e0 50     ld ($ff00+$50), a
```

Game ROM

```
0100 00        nop
0101 c3 50 01  jp $0150
0104           ...
   .           ...
   .           ...
   .           ...
   .           ...
```

Game ROM

```
   .              ...
   .              ...
   .              ...
   .              ...
   .              ...
   .              ...
   .              ...
   .              ...
```

Game ROM

```
0100 00          nop
0101 c3 50 01    jp $0150
0104              ...
   .              ...
   .              ...
   .              ...
   .              ...
```

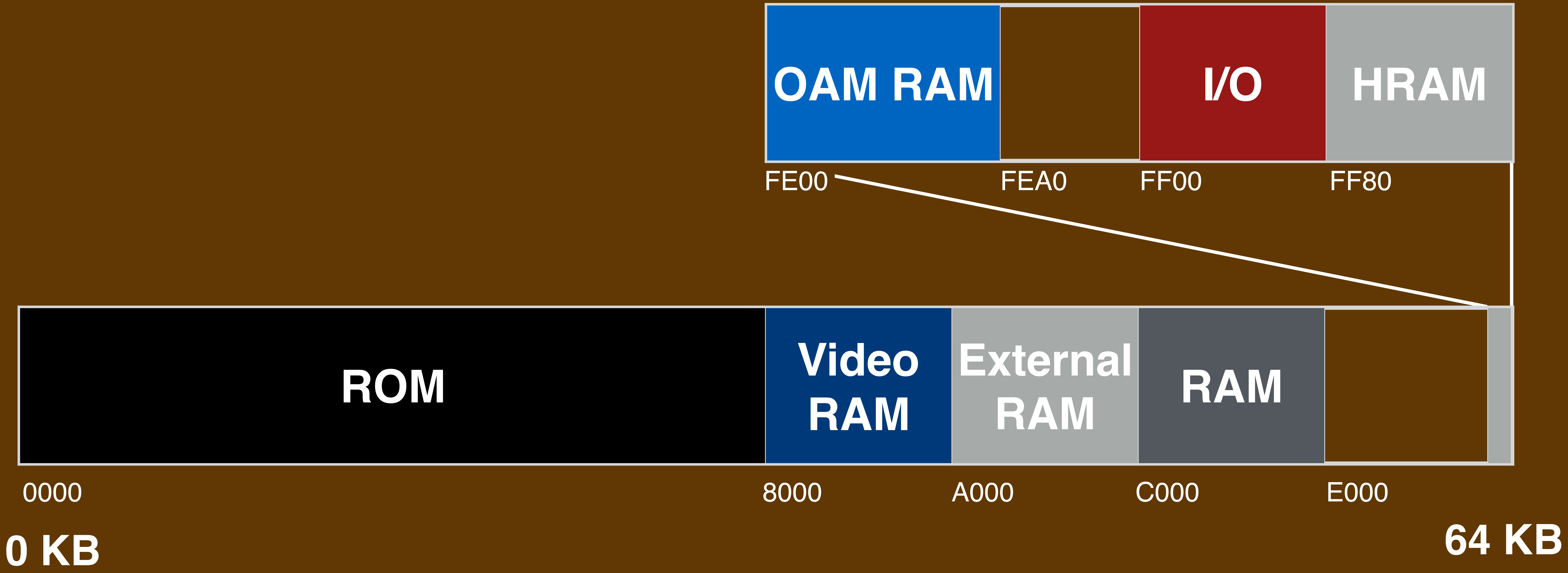
Game ROM

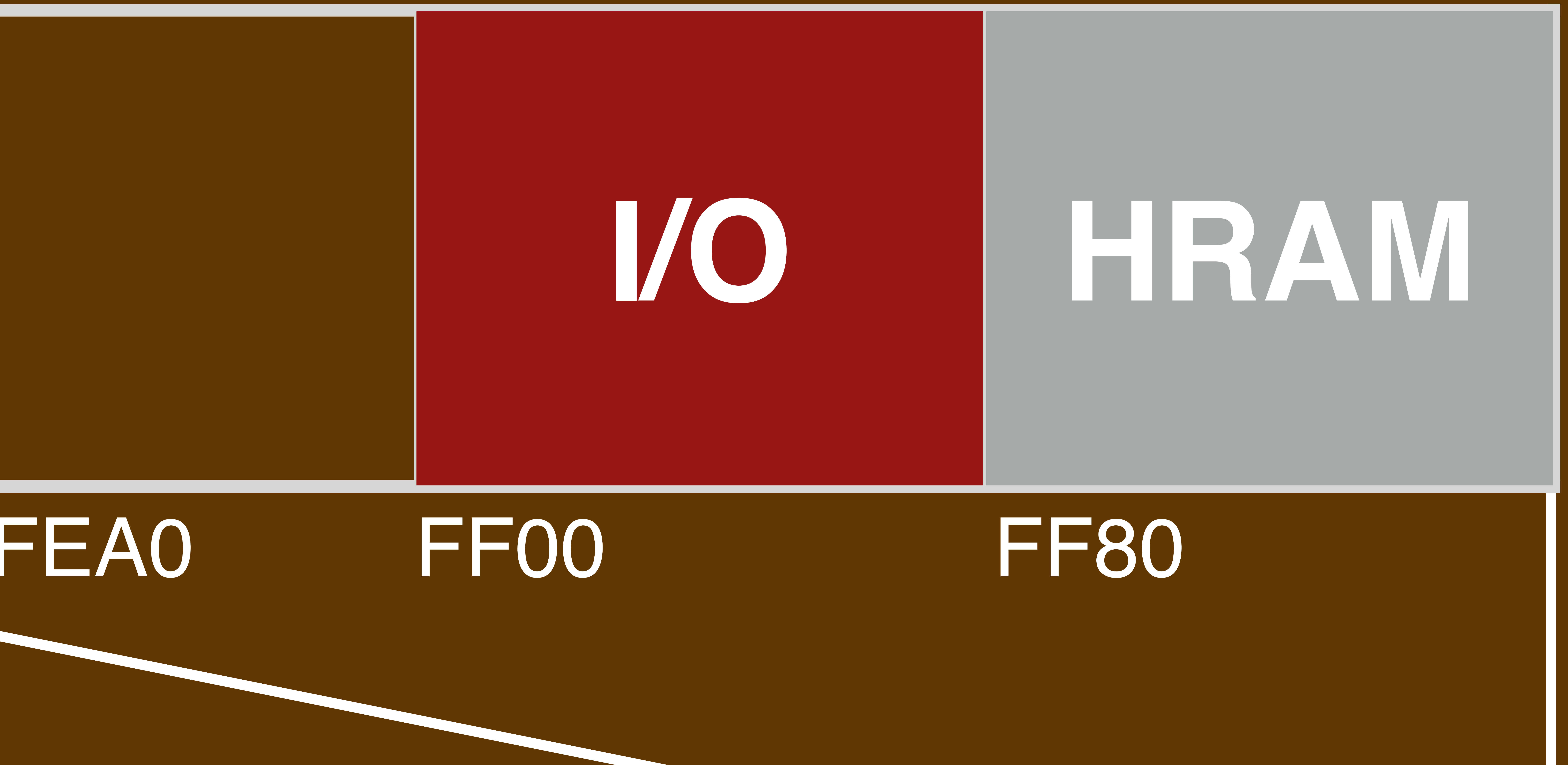
```
   .          ...
   .          ...
   .          ...
   .          ...
   .          ...
   .          ...
   .          ...
   .          ...
```

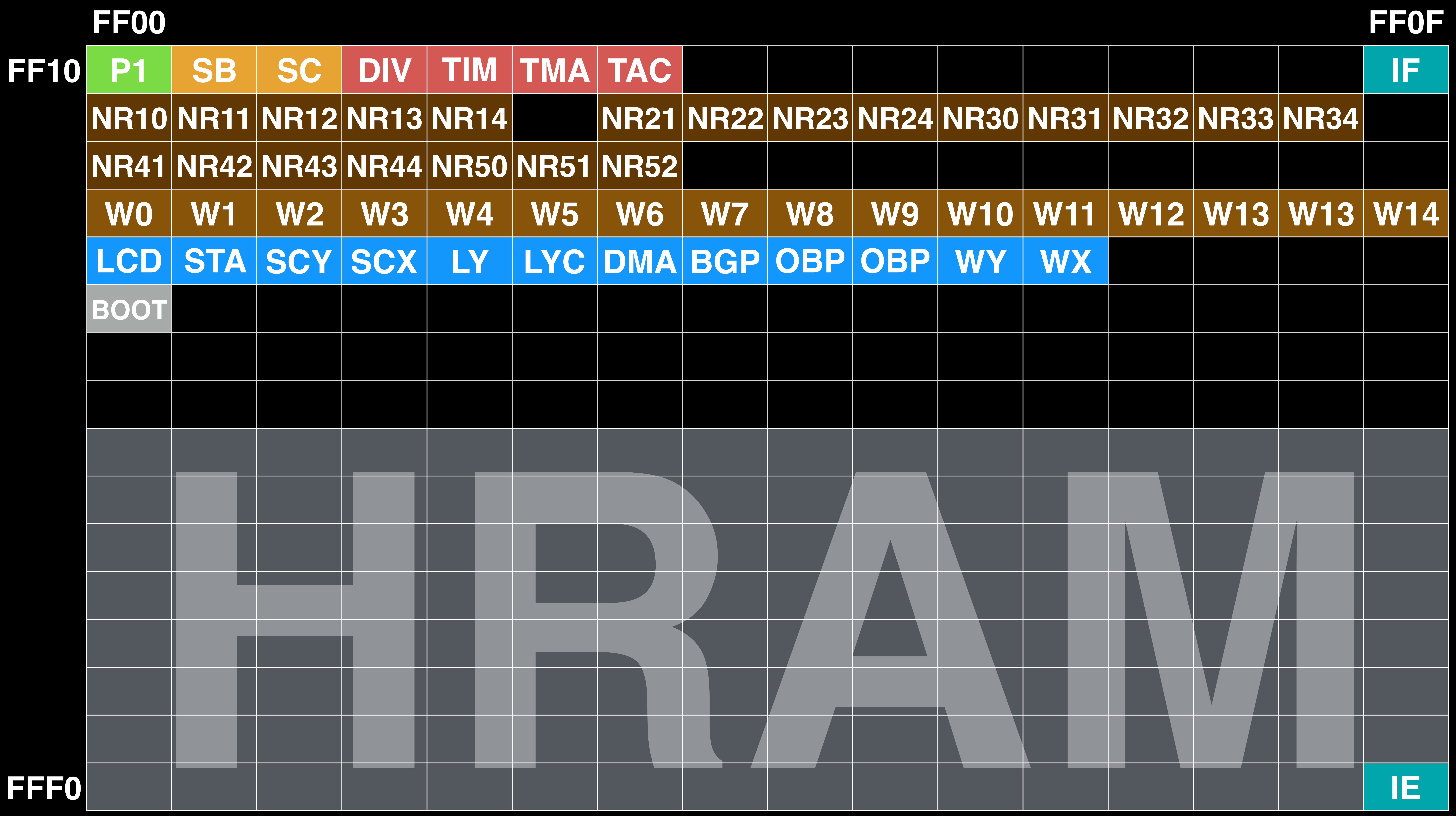
Game ROM

```
0100 00          | nop
0101 c3 50 01    | jp $0150
0104               ...
   .               ...
   .               ...
   .               ...
   .               ...
```

ROM Header		
0100 - 0103	Entry Point	`nop, jp $0150`
0104 - 0133	Nintendo Logo	`$CE, $ED, $66, $66, $CC, ...`
0134 - 0143	Title	“SUPER MARIOLAND”
013F - 0142	Manufacturer Code	`$00`
0143	CGB Flag	`$00`
0144 - 0145	New Licensee Code	`$00, $00`
0146	SGB Flag	`$00`
0147	Cartridge Type	`$01`
0148	ROM Size	`$01`
0149	RAM Size	`$00`
014A	Destination Code	`$00`
014B	Old Licensee Code	`$01`
014C	Mask ROM Version number	`$00`
014D	Header Checksum	`$9E`
014E - 014F	Global Checksum	`$41, $6B`


OAM RAM
I/O
HRAM
FE00
FEA0
FF00
FF80
ROM
Video RAM
External RAM
RAM
0000
8000
A000
C000
E000
0 KB
64 KB


I/O
HRAM
FEA0
FF00
FF80


FF00
FF0F
FF10
P1 SB SC DIV TIM TMA TAC
IF
NR10 NR11 NR12 NR13 NR14
NR21 NR22 NR23 NR24 NR30 NR31 NR32 NR33 NR34
NR41 NR42 NR43 NR44 NR50 NR51 NR52
W0 W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W13 W14
LCD STA SCY SCX LY LYC DMA BGP OBP OBP WY WX
BOOT
HRAM
FFF0
IE

Interrupt Controller

FF0F	IF	Interrupt Flag (R/W)
FFFF	IE	Interrupt Enable (R/W)

Sound Controller

FF10	NR10	Channel 1 Sweep register (R/W)
FF11	NR11	Channel 1 Sound length/Wave pattern duty (R/W)
FF12	NR12	Channel 1 Volume Envelope (R/W)
FF13	NR13	Channel 1 Frequency lo (Write Only)
FF14	NR14	Channel 1 Frequency hi (R/W)
FF16	NR21	Channel 2 Sound Length/Wave Pattern Duty (R/W)
FF17	NR22	Channel 2 Volume Envelope (R/W)
FF18	NR23	Channel 2 Frequency lo data (W)
FF19	NR24	Channel 2 Frequency hi data (R/W)
FF1A	NR30	Channel 3 Sound on/off (R/W)
FF1B	NR31	Channel 3 Sound Length
FF1C	NR32	Channel 3 Select output level (R/W)
FF1D	NR33	Channel 3 Frequency's lower data (W)
FF1E	NR34	Channel 3 Frequency's higher data (R/W)
FF20	NR41	Channel 4 Sound Length (R/W)
FF21	NR42	Channel 4 Volume Envelope (R/W)
FF22	NR43	Channel 4 Polynomial Counter (R/W)
FF23	NR44	Channel 4 Counter/consecutive; Inital (R/W)
FF24	NR50	Channel control / ON-OFF / Volume (R/W)
FF25	NR51	Selection of Sound output terminal (R/W)
FF26	NR52	Sound on/off

Joypad Input

FF00	P1	Joypad (R/W)

Serial Data Transfer

FF01	SB	Serial transfer data (R/W)
FF02	SC	Serial Transfer Control (R/W)

Timer

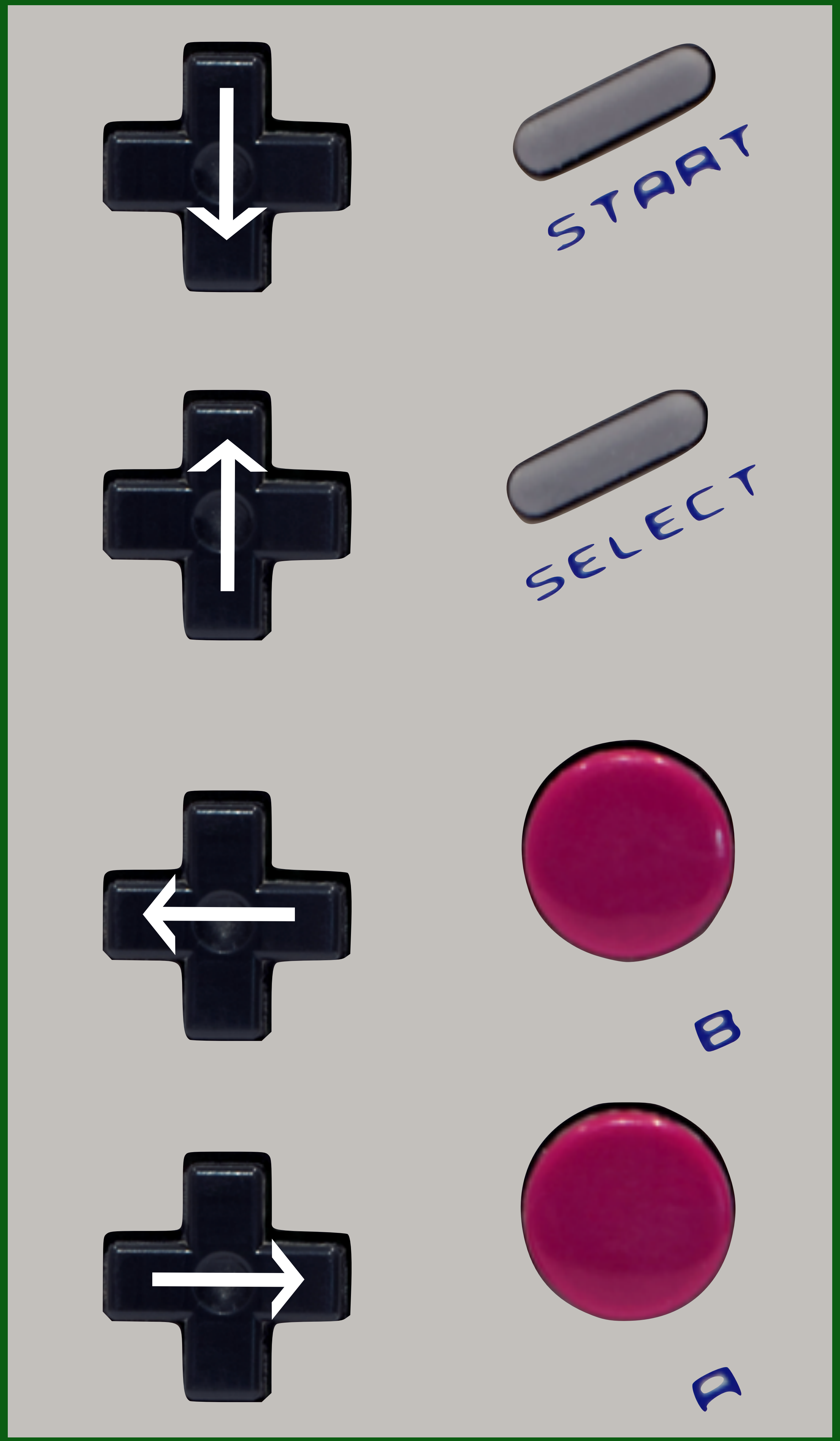
FF04	DIV	Divider Register (R/W)
FF05	TIMA	Timer counter (R/W)
FF06	TMA	Timer Modulo (R/W)
FF07	TAC	Timer Control (R/W)

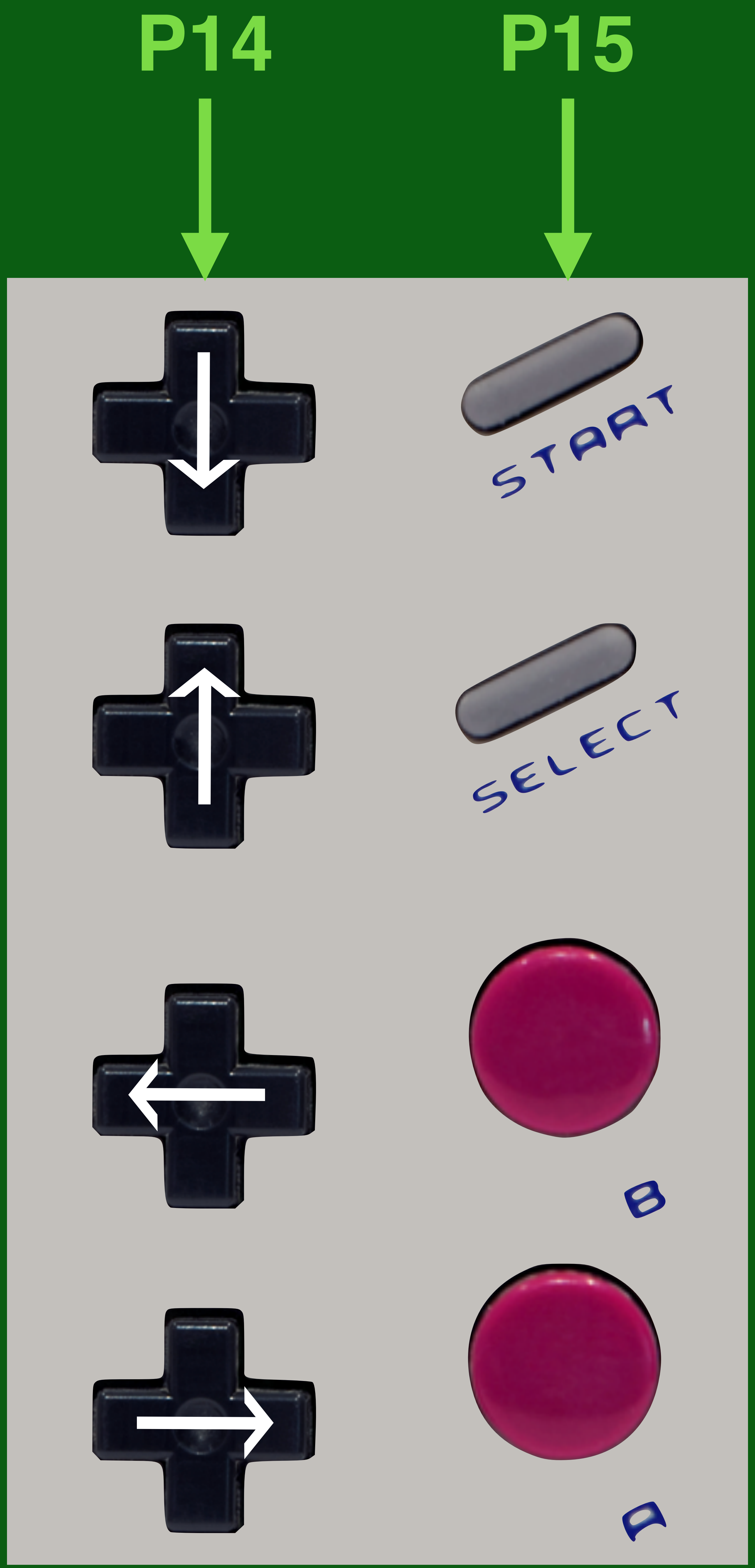
Pixel Processing Unit

FF40	LCDC	LCD Control (R/W)
FF41	STAT	LCDC Status (R/W)
FF42	SCY	Scroll Y (R/W)
FF43	SCX	Scroll X (R/W)
FF44	LY	LCDC Y-Coordinate (R)
FF45	LYC	LY Compare (R/W)
FF46	DMA	DMA Transfer and Start Address (W)
FF47	BGP	BG Palette Data (R/W)
FF48	OBP0	Object Palette 0 Data (R/W)
FF49	OBP1	Object Palette 1 Data (R/W)
FF4A	WY	Window Y Position (R/W)
FF4B	WX	Window X Position minus 7 (R/W)

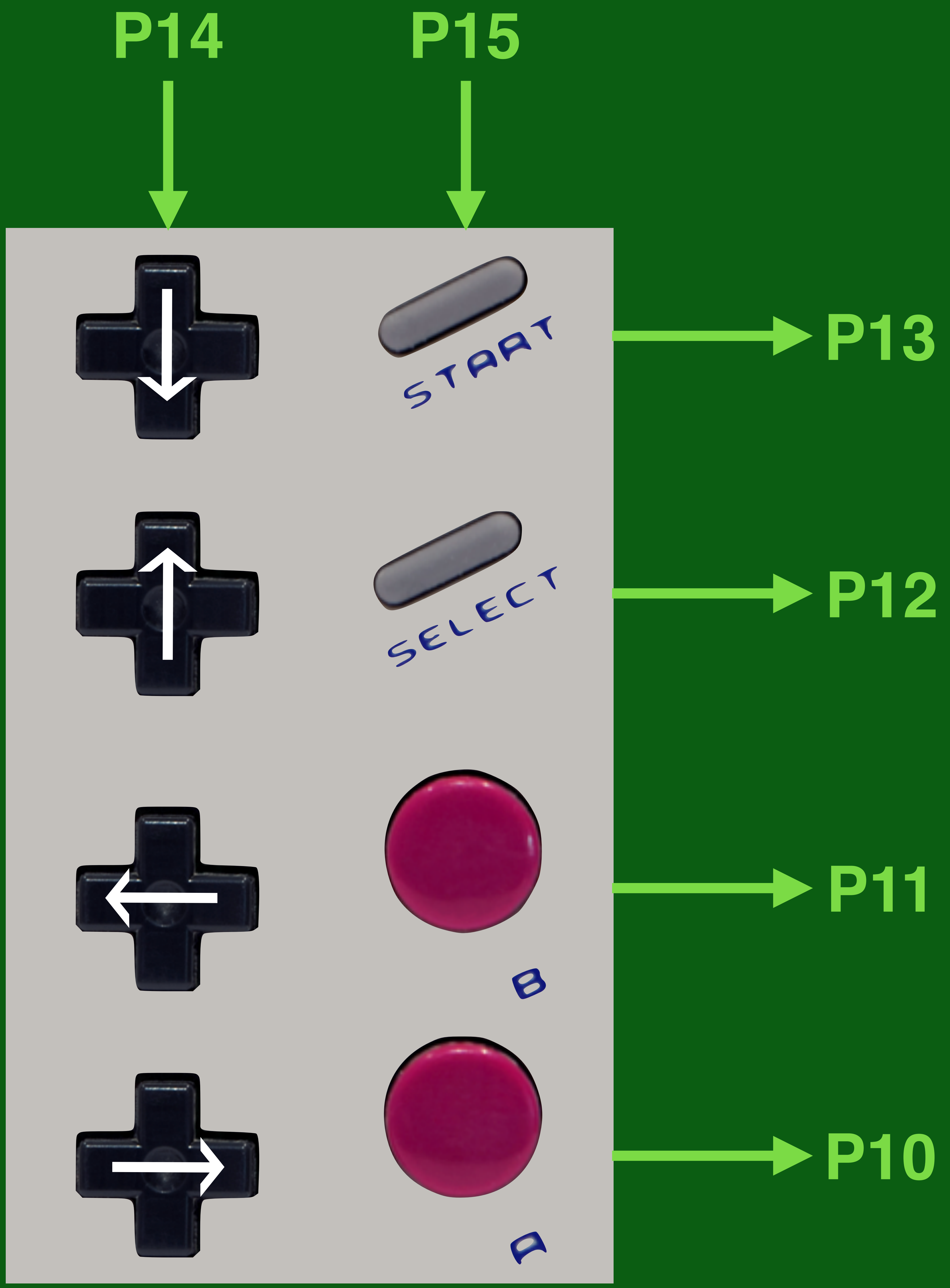
Joypad Input


OFF · ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES


START
SELECT
B
A


P14
P15
START
SELECT
B
A

Joypad Input		
FF00	**P1**	**Joypad (R/W)**
5	**P15**	Select Button Keys
4	**P14**	Select Direction Keys
3	**P13**	Input Down / Start
2	**P12**	Input Up / Select
1	**P11**	Input Left / Button B
0	**P10**	Input Right / Button A



Serial Data Transfer


OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES


OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES


DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES
data →
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES


data →
data ←
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES

data →
data ←
clock ⇆
OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
A
B
SELECT
START
PHONES
OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
A
B
SELECT
START
PHONES

Serial Data Transfer		
FF01	**SB**	**Serial transfer data (R/W)**
FF02	**SC**	**Serial Transfer Control (R/W)**
7		Transfer Start Flag
0		Clock Source (internal/external)

data →

data ←

clock ⇆

	clock speeds
out	8,192 kHz
in	0 kHz – 524,288 kHz

Serial Data Transfer		
FF01	**SB**	**Serial transfer data (R/W)**
FF02	**SC**	**Serial Transfer Control (R/W)**
7		Transfer Start Flag
0		Clock Source (internal/external)

Timer

Timer		
FF04	**DIV**	**Divider Register (R/W)**
FF05	**TIMA**	**Timer counter (R/W)**
FF06	**TMA**	**Timer Modulo (R/W)**
FF07	**TAC**	**Timer Control (R/W)**
2		Timer Start
1-0		Input Clock Select

clock speeds		
01	262,144 Hz	4
10	65,536 Hz	16
11	16,384 Hz	64
00	4,096 Hz	256

Interrupt Controller

Interrupt Controller		
FFFF	**IE**	**Interrupt Enable (R/W)**
4		Joypad
3		Serial
2		Timer
1		LCD STAT
0		V-Blank

Interrupt Controller					
FFFF	**IE**	**Interrupt Enable (R/W)**	**FF0F**	**IF**	**Interrupt Flag (R/W)**
4		Joypad	4		Joypad
3		Serial	3		Serial
2		Timer	2		Timer
1		LCD STAT	1		LCD STAT
0		V-Blank	0		V-Blank

Interprupt Controller					
FFFF	**IE**	**Interrupt Enable (R/W)**	**FF0F**	**IF**	**Interrupt Flag (R/W)**
4		Joypad	4		Joypad **RST $60**
3		Serial	3		Serial **RST $58**
2		Timer	2		Timer **RST $50**
1		LCD STAT	1		LCD STAT **RST $48**
0		V-Blank	0		V-Blank **RST $40**

Sound Controller

Sound Controller		
FF10	NR10	Channel 1 Sweep register (R/W)
FF11	NR11	Channel 1 Sound length/Wave pattern duty (R/W)
FF12	NR12	Channel 1 Volume Envelope (R/W)
FF13	NR13	Channel 1 Frequency lo (Write Only)
FF14	NR14	Channel 1 Frequency hi (R/W)
FF16	NR21	Channel 2 Sound Length/Wave Pattern Duty (R/W)
FF17	NR22	Channel 2 Volume Envelope (R/W)
FF18	NR23	Channel 2 Frequency lo data (W)
FF19	NR24	Channel 2 Frequency hi data (R/W)
FF1A	NR30	Channel 3 Sound on/off (R/W)
FF1B	NR31	Channel 3 Sound Length
FF1C	NR32	Channel 3 Select output level (R/W)
FF1D	NR33	Channel 3 Frequency's lower data (W)
FF1E	NR34	Channel 3 Frequency's higher data (R/W)
FF20	NR41	Channel 4 Sound Length (R/W)
FF21	NR42	Channel 4 Volume Envelope (R/W)
FF22	NR43	Channel 4 Polynomial Counter (R/W)
FF23	NR44	Channel 4 Counter/consecutive; Inital (R/W)
FF24	NR50	Channel control / ON-OFF / Volume (R/W)
FF25	NR51	Selection of Sound output terminal (R/W)
FF26	NR52	Sound on/off

Voice 1

Voice 2

Voice 3

Voice 4

	4	3	2	1	0
Voice 1	NR14	NR13	NR12	NR11	NR10
Voice 2	NR24	NR23	NR22	NR21	NR20
Voice 3	NR34	NR33	NR32	NR31	NR30
Voice 4	NR44	NR43	NR42	NR41	NR40

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Voice 1	NR14	NR13	NR12	NR11	NR10
Voice 2	NR24	NR23	NR22	NR21	NR20
Voice 3	NR34	NR33	NR32	NR31	NR30
Voice 4	NR44	NR43	NR42	NR41	NR40

	Control 4	Frequency 3	Volume 2	Length 1	Sweep 0
Pulse A					
Pulse B					
Wave					
Noise					

	Control							
	4							
Pulse A	T	L				F	F	F
Pulse B	T	L				F	F	F
Wave	T	L				F	F	F
Noise	T	L						

	Frequency							
	3							
Pulse A	F	F	F	F	F	F	F	F
Pulse B	F	F	F	F	F	F	F	F
Wave	F	F	F	F	F	F	F	F
Noise	S	S	S	S	W	D	D	D

	Volume							
	2							
Pulse A	V	V	V	V	A	P	P	P
Pulse B	V	V	V	V	A	P	P	P
Wave		V	V					
Noise	V	V	V	V	A	P	P	P

	Length							
	1							
Pulse A	D	D	L	L	L	L	L	L
Pulse B	D	D	L	L	L	L	L	L
Wave	L	L	L	L	L	L	L	L
Noise			L	L	L	L	L	L

	Sweep							
	0							
Pulse A		P	P	P	N	S	S	S
Pulse B								
Wave								
Noise								

Control
4

Pulse A	T	L				F	F	F
Pulse B	T	L				F	F	F
Wave	T	L				F	F	F
Noise	T	L						

Frequency
3

Pulse A	F	F	F	F	F	F	F	F
Pulse B	F	F	F	F	F	F	F	F
Wave	F	F	F	F	F	F	F	F
Noise	S	S	S	S	W	D	D	D

Volume
2

Pulse A	V	V	V	V	A	P	P	P
Pulse B	V	V	V	V	A	P	P	P
Wave		V	V					
Noise	V	V	V	V	A	P	P	P

Length
1

Pulse A	D	D	L	L	L	L	L	L
Pulse B	D	D	L	L	L	L	L	L
Wave	L	L	L	L	L	L	L	L
Noise			L	L	L	L	L	L

Sweep
0

Pulse A		P	P	P	N	S	S	S
Pulse B								
Wave								
Noise								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								
Wave	T	L				F	F	F	F	F	F	F	F	F	F	F		V	V						L	L	L	L	L	L	L	L								
Noise	T	L							S	S	S	S	W	D	D	D	V	V	V	V	A	P	P	P			L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Wave	T	L				F	F	F	F	F	F	F	F	F	F	F		V	V						L	L	L	L	L	L	L	L								

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Wave	T L _ _ _ F F F	F F F F F F F F	_ V V _ _ _ _ _	L L L L L L L L	_ _ _ _ _ _ _ _

FF30	**W0**	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**	**W8**	**W9**	**W10**	**W11**	**W12**	**W13**	**W13**	**W14**

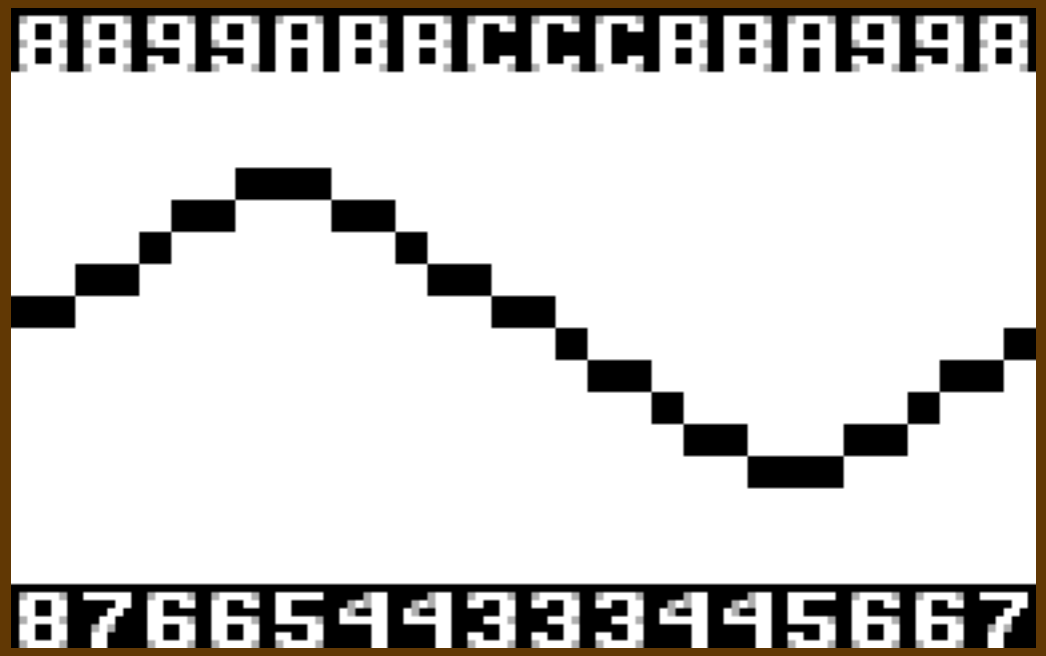
	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0

Wave	T	L				F	F	F	F	F	F	F	F	F	F	F		V	V						L	L	L	L	L	L	L	L								

FF30	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W13	W14

Audio samples by @_drmr_

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Wave	T L _ _ _ F F F	F F F F F F F F	_ V V _ _ _ _ _	L L L L L L L L	_ _ _ _ _ _ _ _



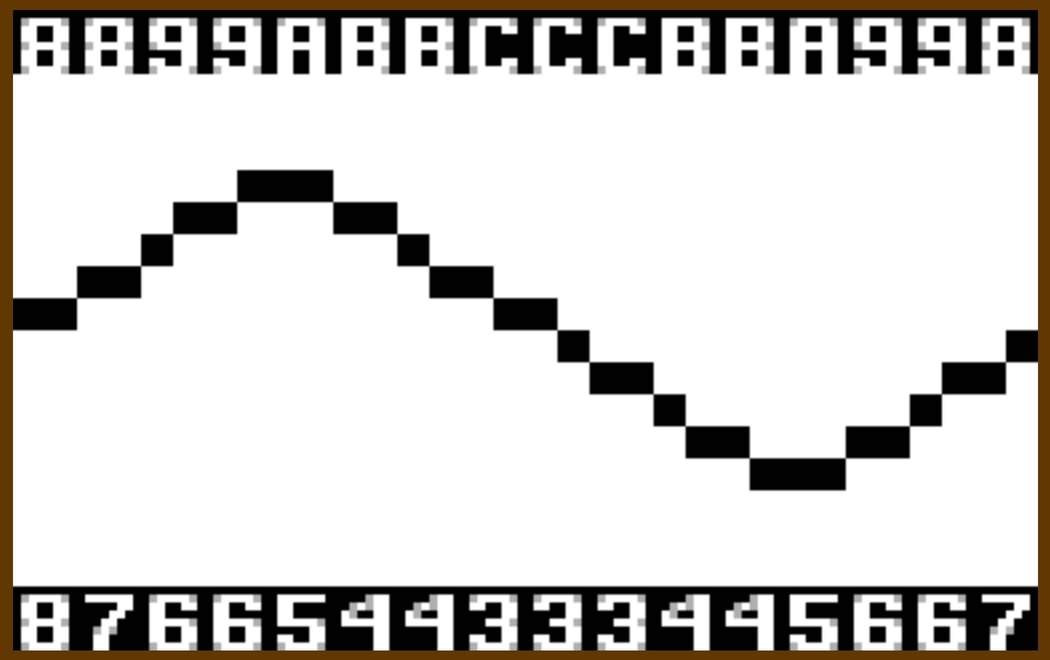
FF30	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W13	W14

Audio samples by @_drmr_

Control	Frequency	Volume	Length	Sweep
4	3	2	1	0

Wave

T	L				F	F	F
F	F	F	F	F	F	F	F
	V	V					
L	L	L	L	L	L	L	L



FF30	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W13	W14

Audio samples by @_drmr_

Control	Frequency	Volume	Length	Sweep
4	3	2	1	0

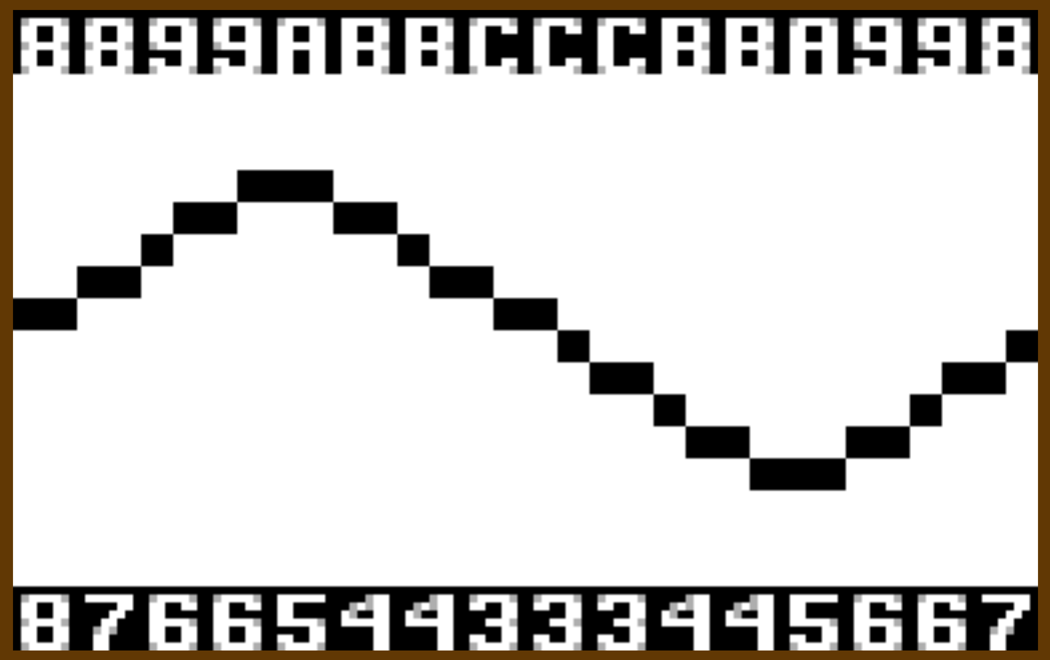
Wave | T | L | | | | F | F | F | | F | F | F | F | F | F | F | F | | V | V | | | | | | L | L | L | L | L | L | L | L | | | | | | | | |

FF30	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W13	W14

Audio samples by @_drmr_

Control	Frequency	Volume	Length	Sweep
4	3	2	1	0

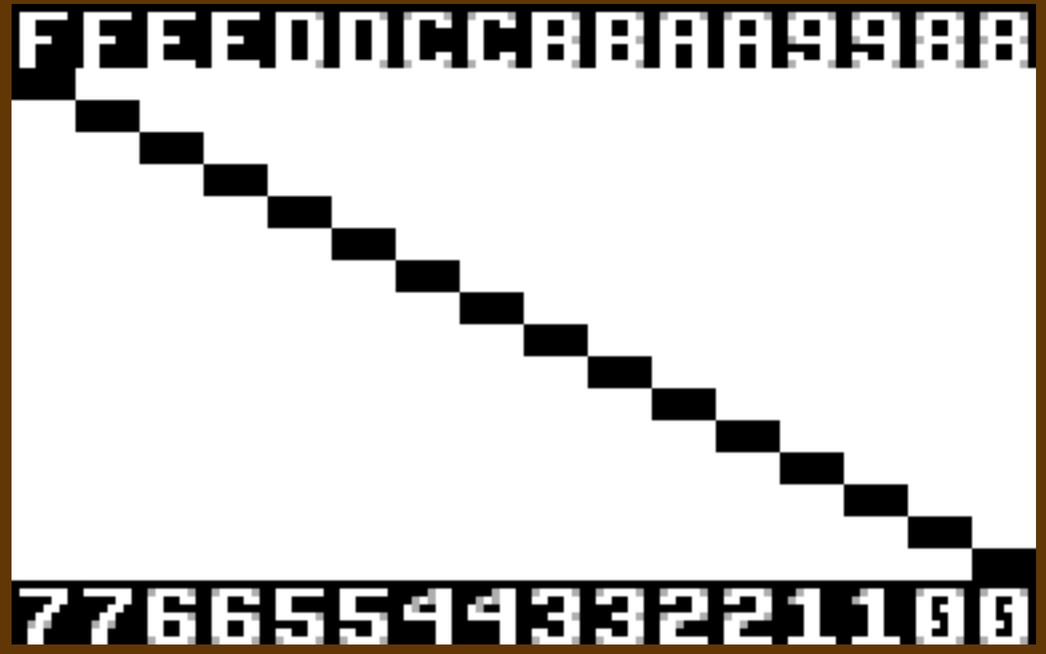
Wave	T L _ _ _ F F F	F F F F F F F F	_ V V _ _ _ _ _	L L L L L L L L	_ _ _ _ _ _ _ _

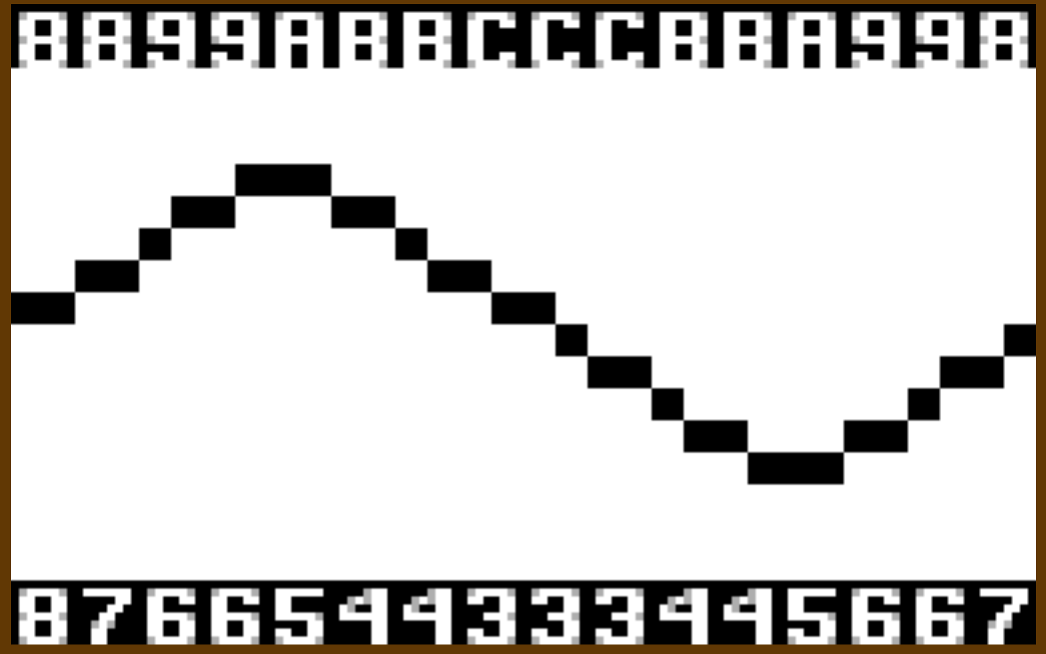


FF30	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W13	W14

Audio samples by @_drmr_

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Wave	T L _ _ _ F F F	F F F F F F F F	_ V V _ _ _ _ _	L L L L L L L L	_ _ _ _ _ _ _ _





FF30	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W13	W14

Audio samples by @_drmr_

	Control 4							
Pulse A	T	L				F	F	F
Pulse B	T	L				F	F	F

	Frequency 3							
Pulse A	F	F	F	F	F	F	F	F
Pulse B	F	F	F	F	F	F	F	F

	Volume 2							
Pulse A	V	V	V	V	A	P	P	P
Pulse B	V	V	V	V	A	P	P	P

	Length 1							
Pulse A	D	D	L	L	L	L	L	L
Pulse B	D	D	L	L	L	L	L	L

	Sweep 0							
Pulse A		P	P	P	N	S	S	S
Pulse B								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

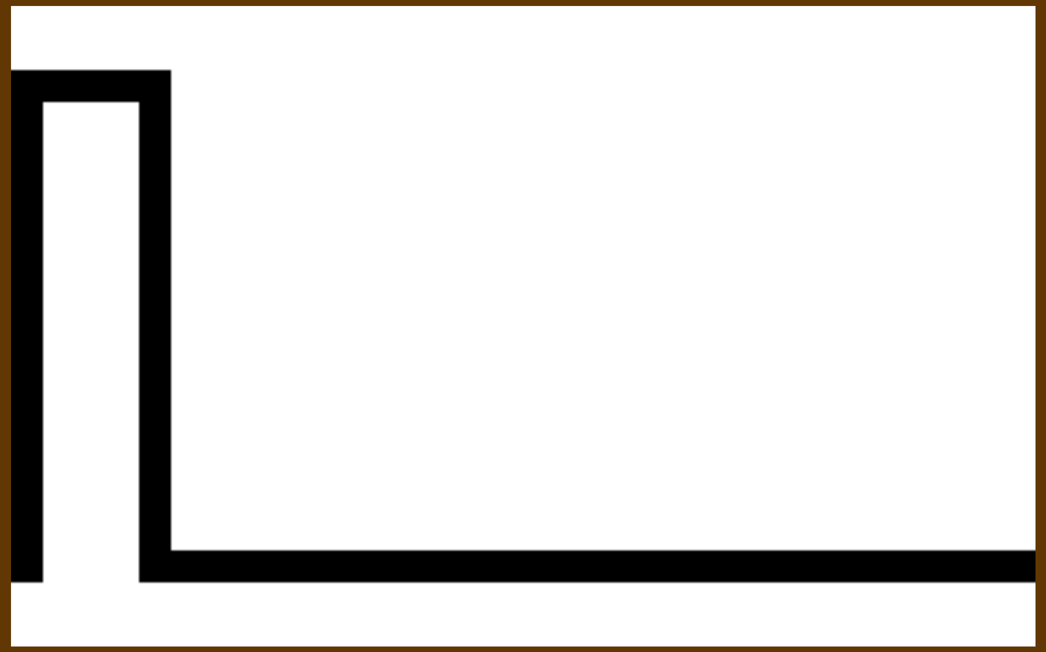
	Control 4							
Pulse A	T	L				F	F	F
Pulse B	T	L				F	F	F

	Frequency 3							
Pulse A	F	F	F	F	F	F	F	F
Pulse B	F	F	F	F	F	F	F	F

	Volume 2							
Pulse A	V	V	V	V	A	P	P	P
Pulse B	V	V	V	V	A	P	P	P

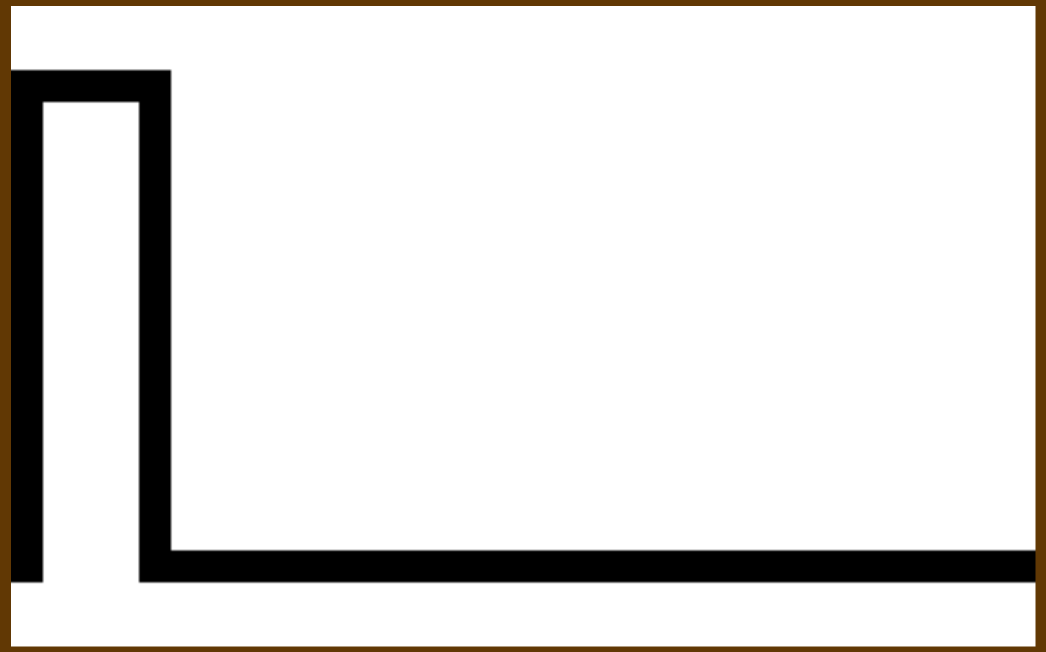
	Length 1							
Pulse A	D	D	L	L	L	L	L	L
Pulse B	D	D	L	L	L	L	L	L

	Sweep 0							
Pulse A		P	P	P	N	S	S	S
Pulse B								

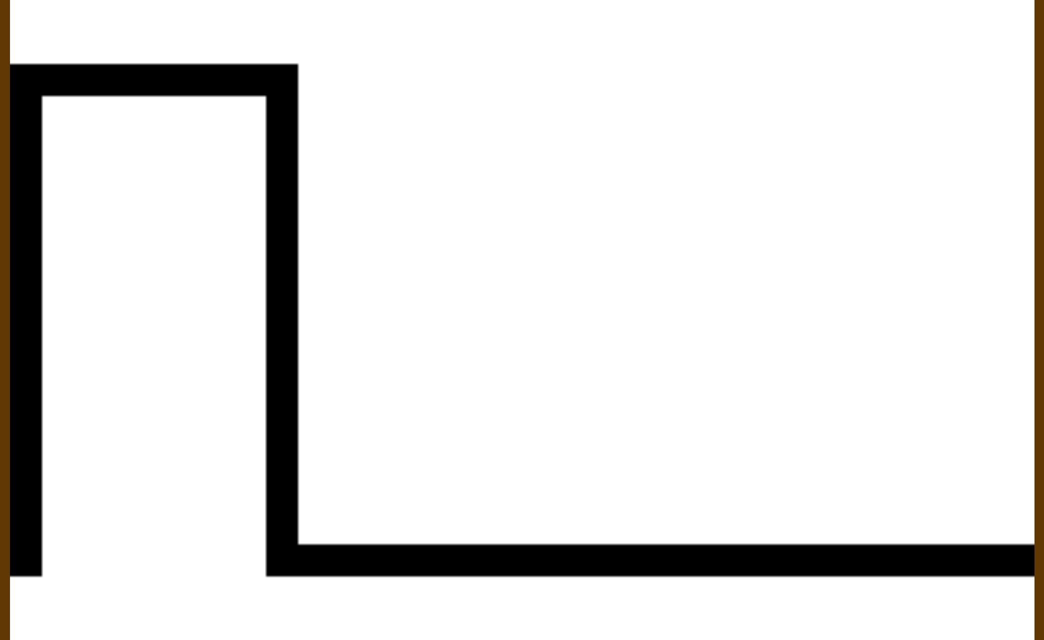


12.5%

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

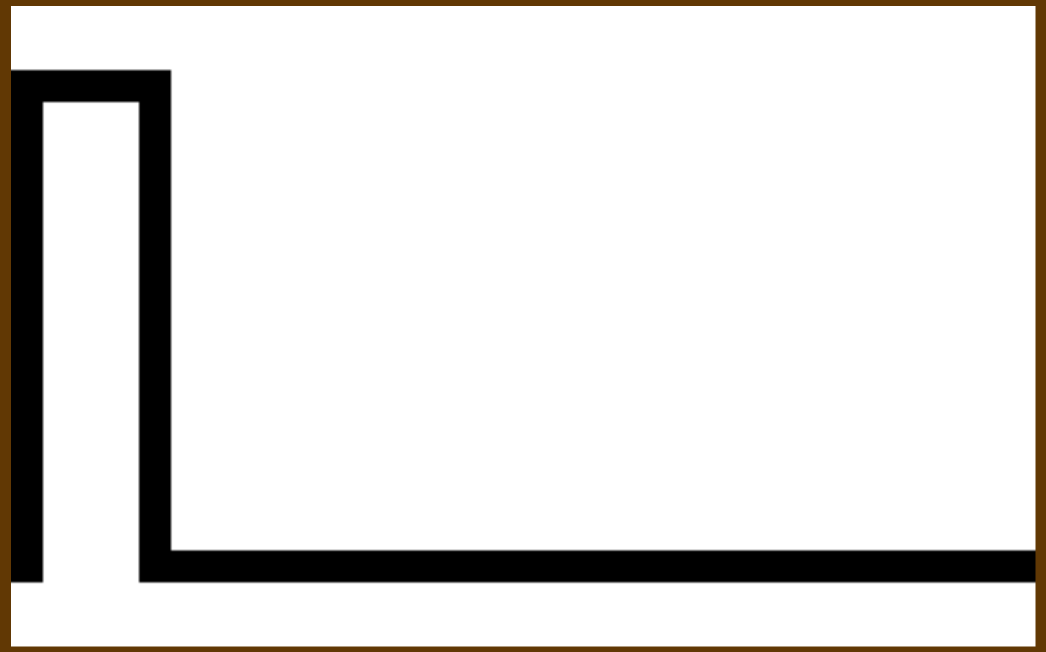


12.5%

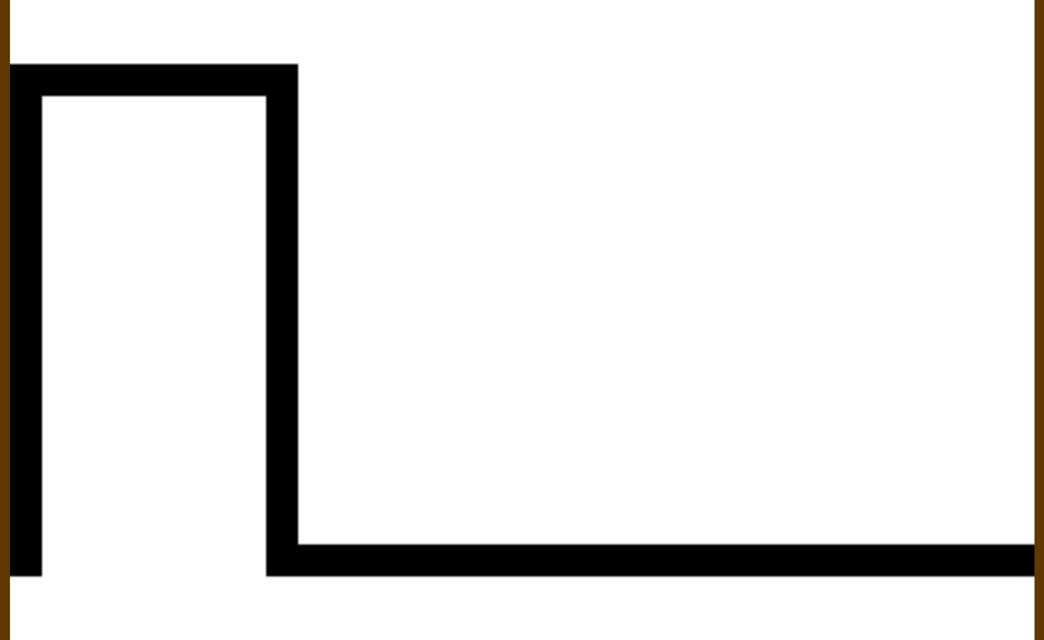


25%

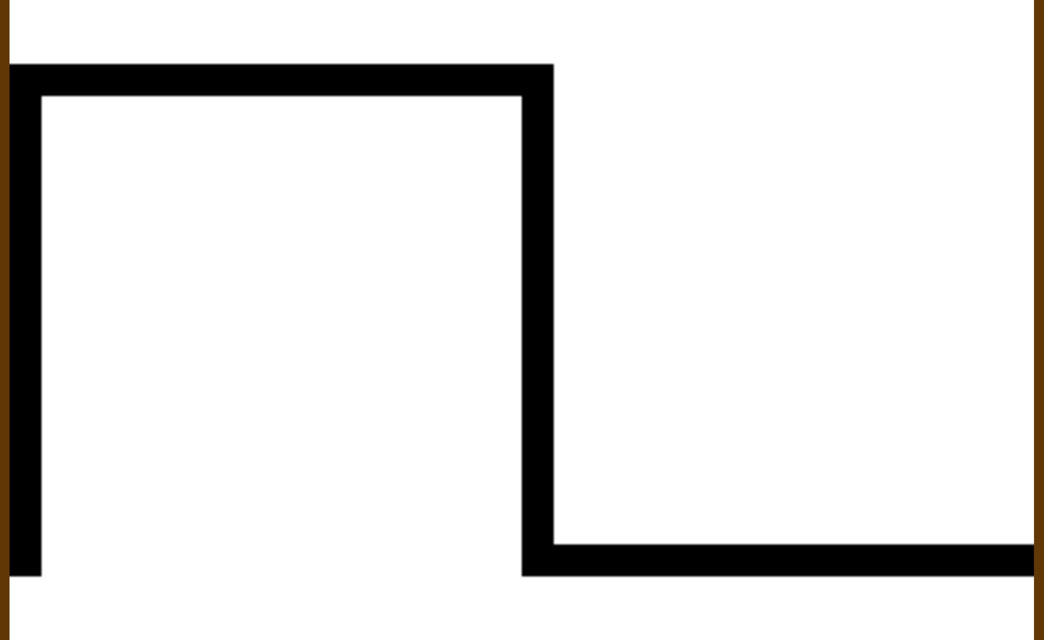
	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								



12.5%

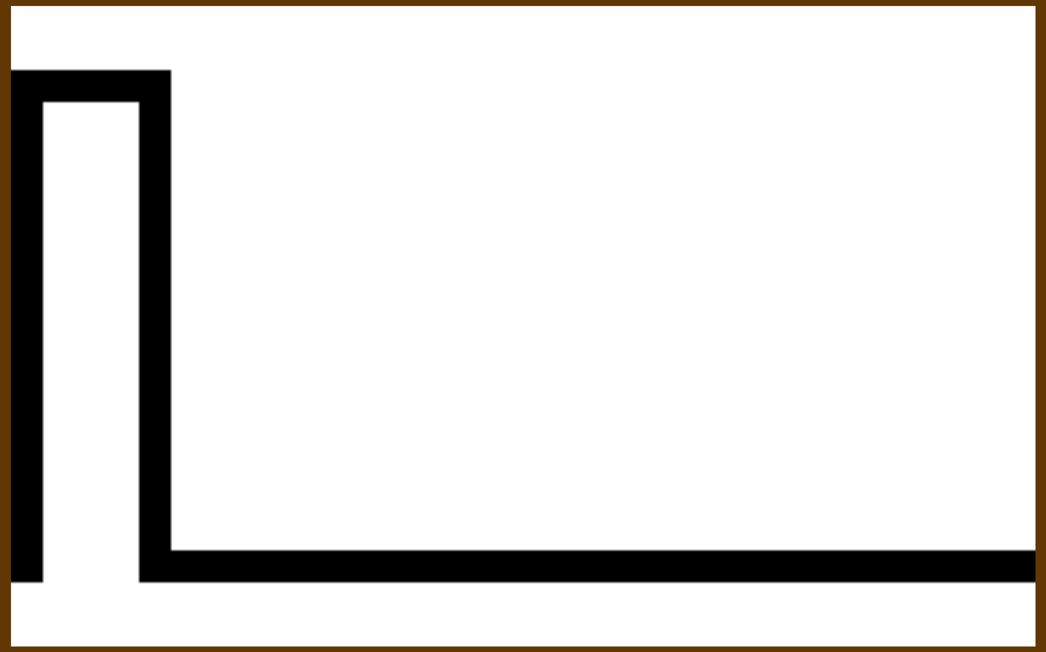


25%

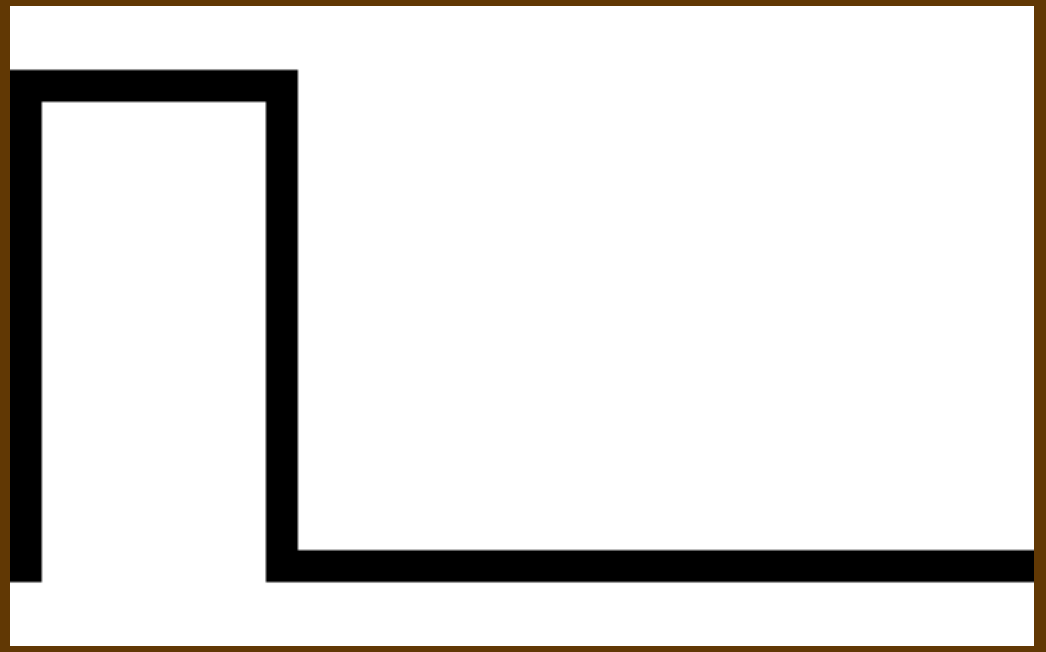


50%

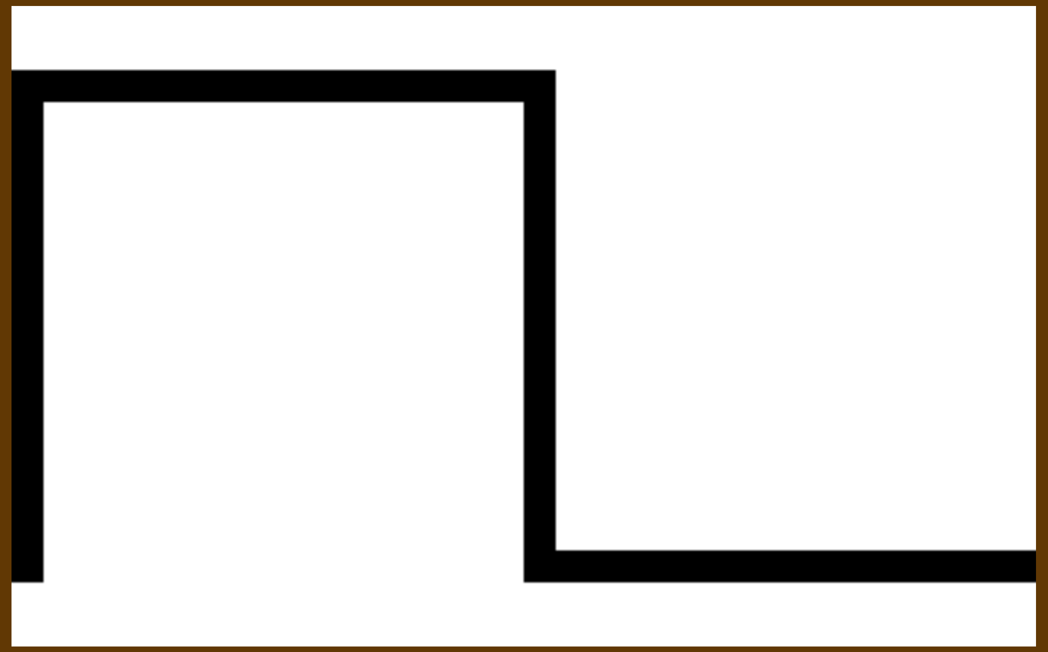
	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Pulse A	T L _ _ _ F F F	F F F F F F F F	V V V V A P P P	D D L L L L L L	_ P P P N S S S
Pulse B	T L _ _ _ F F F	F F F F F F F F	V V V V A P P P	D D L L L L L L	_ _ _ _ _ _ _ _



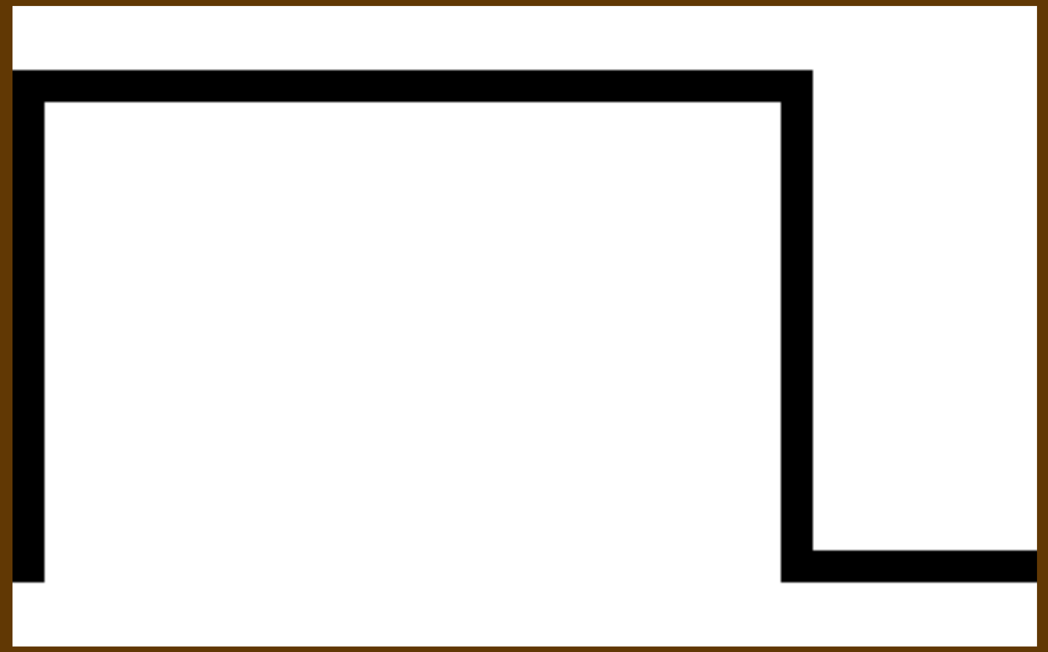
12.5%



25%



50%



75%

	Control							
	4							
Pulse A	T	L				F	F	F
Pulse B	T	L				F	F	F

	Frequency							
	3							
Pulse A	F	F	F	F	F	F	F	F
Pulse B	F	F	F	F	F	F	F	F

	Volume							
	2							
Pulse A	V	V	V	V	A	P	P	P
Pulse B	V	V	V	V	A	P	P	P

	Length							
	1							
Pulse A	D	D	L	L	L	L	L	L
Pulse B	D	D	L	L	L	L	L	L

	Sweep							
	0							
Pulse A		P	P	P	N	S	S	S
Pulse B								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Noise	T L _ _ _ _ _ _	S S S S W D D D	V V V V A P P P	_ _ L L L L L L	_ _ _ _ _ _ _ _

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Noise	T L	S S S S W D D D	V V V V A P P P	L L L L L L	

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Noise	T L	S S S S W D D D	V V V V A P P P	L L L L L L	

	Control	Frequency	Volume	Length	Sweep
	4	3	2	1	0
Noise	T L _ _ _ _ _ _	$\boxed{\text{S S S S W D D D}}$	V V V V A P P P	_ _ L L L L L L	_ _ _ _ _ _ _ _

	Control								Frequency								Volume								Length								Sweep							
	4								3								2								1								0							
Pulse A	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L		P	P	P	N	S	S	S
	NR14								NR13								NR12								NR11								NR10							
Pulse B	T	L				F	F	F	F	F	F	F	F	F	F	F	V	V	V	V	A	P	P	P	D	D	L	L	L	L	L	L								
	NR24								NR23								NR22								NR21								NR20							
Wave	T	L				F	F	F	F	F	F	F	F	F	F	F		V	V						L	L	L	L	L	L	L	L								
	NR34								NR33								NR32								NR31								NR30							
Noise	T	L							S	S	S	S	W	D	D	D	V	V	V	V	A	P	P	P			L	L	L	L	L	L								
	NR44								NR43								NR42								NR41								NR40							

Control

4

Pulse A	T	L				F	F	F	NR14
Pulse B	T	L				F	F	F	NR24
Wave	T	L				F	F	F	NR34
Noise	T	L							NR44

Frequency

3

Pulse A	F	F	F	F	F	F	F	F	NR13
Pulse B	F	F	F	F	F	F	F	F	NR23
Wave	F	F	F	F	F	F	F	F	NR33
Noise	S	S	S	S	W	D	D	D	NR43

Volume

2

Pulse A	V	V	V	V	A	P	P	P	NR12
Pulse B	V	V	V	V	A	P	P	P	NR22
Wave		V	V						NR32
Noise	V	V	V	V	A	P	P	P	NR42

Length

1

Pulse A	D	D	L	L	L	L	L	L	NR11
Pulse B	D	D	L	L	L	L	L	L	NR21
Wave	L	L	L	L	L	L	L	L	NR31
Noise			L	L	L	L	L	L	NR41

Sweep

0

Pulse A		P	P	P	N	S	S	S	NR10
Pulse B									NR20
Wave									NR30
Noise									NR40

								NR50
								NR51
								NR52

Channel	Section	Register								
Pulse A	Control 4	NR14	T	L				F	F	F
Pulse A	Frequency 3	NR13	F	F	F	F	F	F	F	F
Pulse A	Volume 2	NR12	V	V	V	V	A	P	P	P
Pulse A	Length 1	NR11	D	D	L	L	L	L	L	L
Pulse A	Sweep 0	NR10		P	P	P	N	S	S	S
Pulse B	Control 4	NR24	T	L				F	F	F
Pulse B	Frequency 3	NR23	F	F	F	F	F	F	F	F
Pulse B	Volume 2	NR22	V	V	V	V	A	P	P	P
Pulse B	Length 1	NR21	D	D	L	L	L	L	L	L
Pulse B	Sweep 0	NR20								
Wave	Control 4	NR34	T	L				F	F	F
Wave	Frequency 3	NR33	F	F	F	F	F	F	F	F
Wave	Volume 2	NR32		V	V					
Wave	Length 1	NR31	L	L	L	L	L	L	L	L
Wave	Sweep 0	NR30								
Noise	Control 4	NR44	T	L						
Noise	Frequency 3	NR43	S	S	S	S	W	D	D	D
Noise	Volume 2	NR42	V	V	V	V	A	P	P	P
Noise	Length 1	NR41			L	L	L	L	L	L
Noise	Sweep 0	NR40								

Register								
NR50	A	L	L	L	B	R	R	R
NR51	N	W	2	1	N	W	2	1
NR52	P				N	W	2	1

Control	Frequency	Volume	Length	Sweep
4	3	2	1	0

Pulse A	T	L				F	F	F	NR14
	F	F	F	F	F	F	F	F	NR13
	V	V	V	V	A	P	P	P	NR12
	D	D	L	L	L	L	L	L	NR11
		P	P	P	N	S	S	S	NR10
Pulse B	T	L				F	F	F	NR24
	F	F	F	F	F	F	F	F	NR23
	V	V	V	V	A	P	P	P	NR22
	D	D	L	L	L	L	L	L	NR21
									NR20
Wave	T	L				F	F	F	NR34
	F	F	F	F	F	F	F	F	NR33
		V	V						NR32
	L	L	L	L	L	L	L	L	NR31
									NR30
Noise	T	L							NR44
	S	S	S	S	W	D	D	D	NR43
	V	V	V	V	A	P	P	P	NR42
			L	L	L	L	L	L	NR41
									NR40
	$\boxed{A}$	L	L	L	$\boxed{B}$	R	R	R	NR50
	N	W	2	1	N	W	2	1	NR51
	P				N	W	2	1	NR52

	Control 4								
Pulse A	NR14	T	L				F	F	F
Pulse B	NR24	T	L				F	F	F
Wave	NR34	T	L				F	F	F
Noise	NR44	T	L						

	Frequency 3								
Pulse A	NR13	F	F	F	F	F	F	F	F
Pulse B	NR23	F	F	F	F	F	F	F	F
Wave	NR33	F	F	F	F	F	F	F	F
Noise	NR43	S	S	S	S	W	D	D	D

	Volume 2								
Pulse A	NR12	V	V	V	V	A	P	P	P
Pulse B	NR22	V	V	V	V	A	P	P	P
Wave	NR32		V	V					
Noise	NR42	V	V	V	V	A	P	P	P

	Length 1								
Pulse A	NR11	D	D	L	L	L	L	L	L
Pulse B	NR21	D	D	L	L	L	L	L	L
Wave	NR31	L	L	L	L	L	L	L	L
Noise	NR41			L	L	L	L	L	L

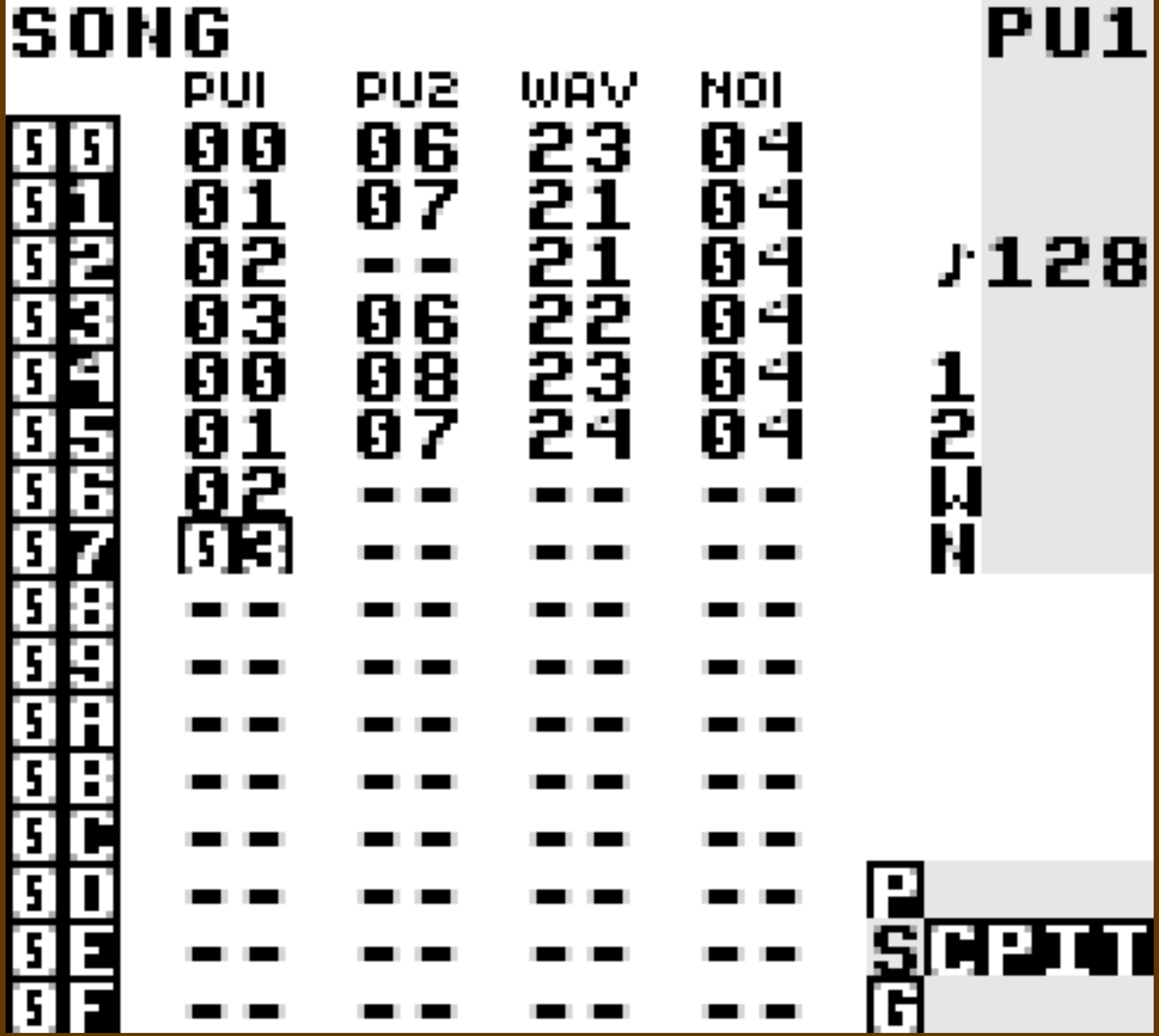
	Sweep 0								
Pulse A	NR10		P	P	P	N	S	S	S
Pulse B	NR20								
Wave	NR30								
Noise	NR40								

NR50	A	L	L	L	B	R	R	R
NR51	N	W	2	1	N	W	2	1
NR52	P				N	W	2	1

	Control 4	Frequency 3	Volume 2	Length 1	Sweep 0
Pulse A	T L F F F (NR14)	F F F F F F F F (NR13)	V V V V A P P P (NR12)	D D L L L L L L (NR11)	P P P N S S S (NR10)
Pulse B	T L F F F (NR24)	F F F F F F F F (NR23)	V V V V A P P P (NR22)	D D L L L L L L (NR21)	(NR20)
Wave	T L F F F (NR34)	F F F F F F F F (NR33)	V V (NR32)	L L L L L L L L (NR31)	(NR30)
Noise	T L (NR44)	S S S S W D D D (NR43)	V V V V A P P P (NR42)	L L L L L L (NR41)	(NR40)

NR50	A L L L B R R R
NR51	N W 2 1 N W 2 1
NR52	P N W 2 1

Little Sound Dj



Toriena & NNNNNNNNNN

Cheapshot - BrainFreeze Abortifacient remix (NNNNNNNNNN remix)

Pixel Processing Unit

Pixel Processing Unit					
FF40	**LCDC**	**LCD Control (R/W)**	**FF42**	**SCY**	**Scroll Y (R/W)**
7		LCD Display Enable	**FF43**	**SCX**	**Scroll X (R/W)**
6		Window Tile Map Address	**FF44**	**LY**	**LCDC Y-Coordinate (R)**
5		Window Enable	**FF45**	**LYC**	**LY Compare (R/W)**
4		BG & Window Tile Data	**FF46**	**DMA**	**DMA Transfer and Start**
3		BG Tile Map Address	**FF47**	**BGP**	**BG Palette (R/W)**
2		OBJ Size	**FF48**	**OBP0**	**Object Palette 0 (R/W)**
1		OBJ Enable	**FF49**	**OBP1**	**Object Palette 1 (R/W)**
0		BG Enable	**FF4A**	**WY**	**Window Y Position (R/W)**
FF41	**STAT**	**LCDC Status (R/W)**	**FF4B**	**WX**	**Window X Position (R/W)**
6		LYC=LY Interrupt			
5		Mode 2 OAM Interrupt			
4		Mode 1 V-Blank Interrupt			
3		Mode 0 H-Blank Interrupt			
2		LYC=LY Flag			
1-0		Mode			


OFF · ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM
B
A
SELECT
START
PHONES


DOT MATRIX WITH STEREO SOUND
BATTERY

Specifications

- 160x144 pixels
- 4 shades of gray
- 8x8 pixel tile-based, 20x18 tiles
- 40 sprites (10 per line)
- 8 KB VRAM

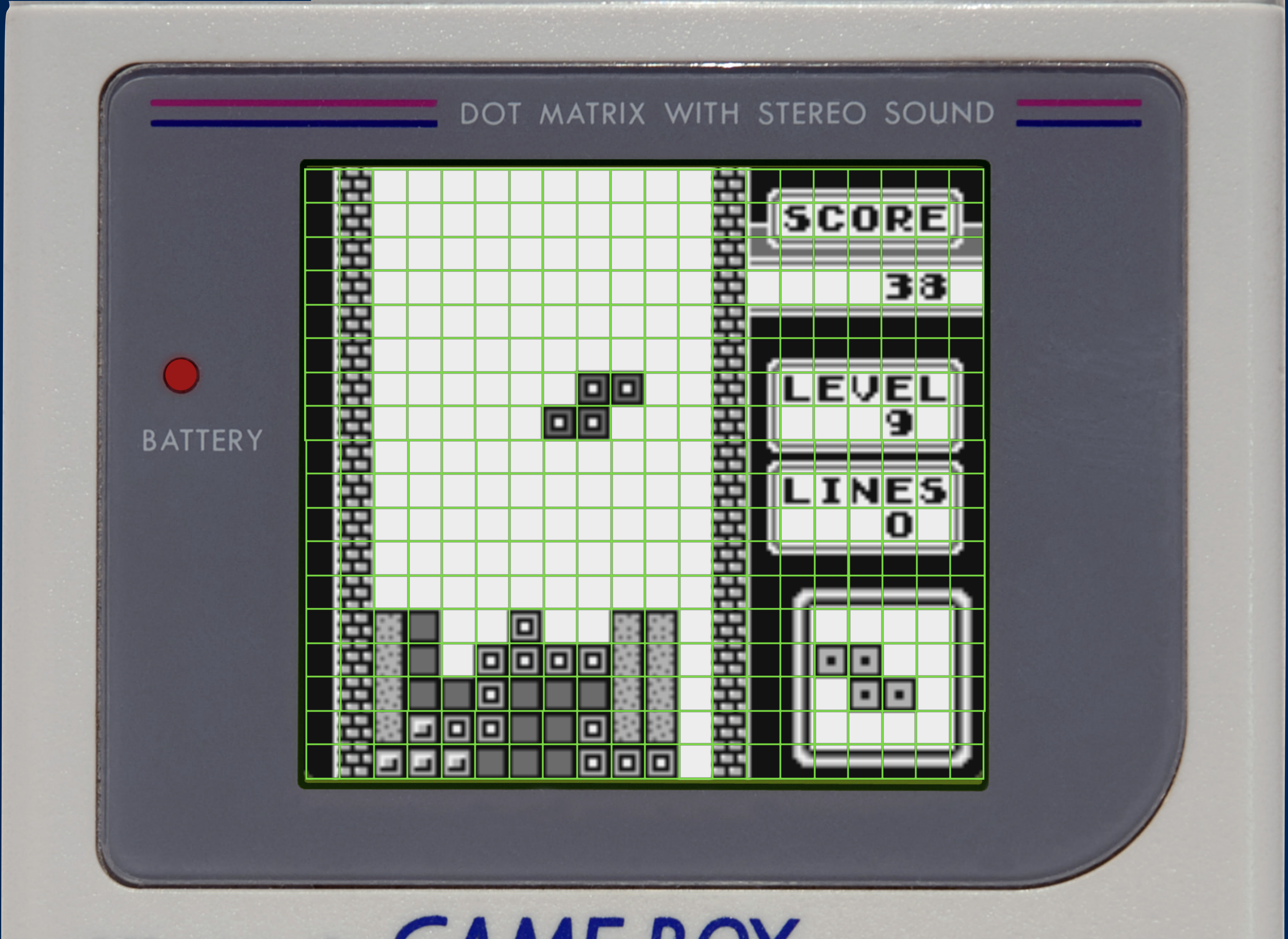
Specifications

- 160x144 pixels
- 4 shades of gray
- 8x8 pixel tile-based, 20x18 tiles
- 40 sprites (10 per line)
- 8 KB VRAM


OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY


OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
SCORE
38
LEVEL
9
LINES
0

20x18 Background Tiles

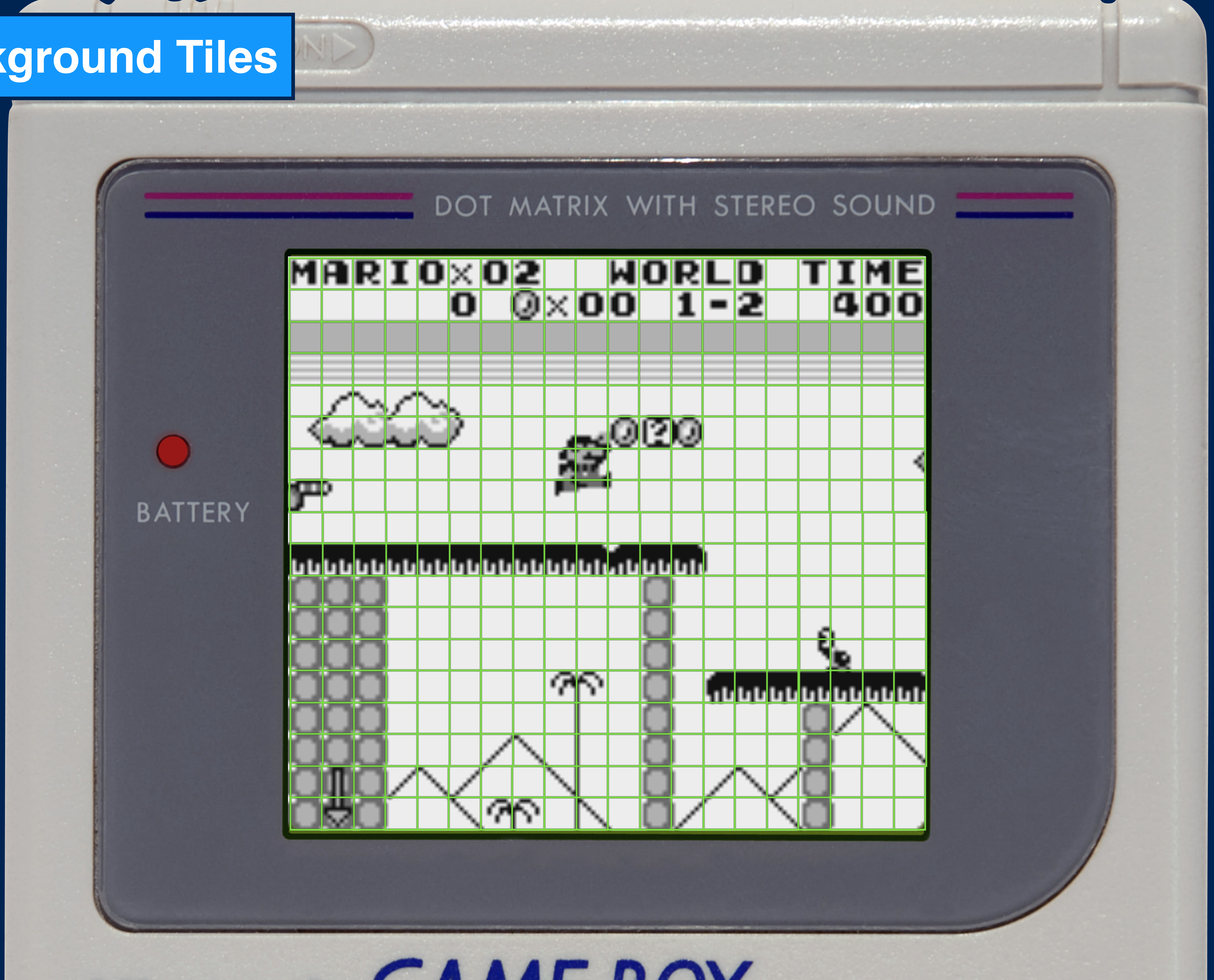


20x18 Background Tiles

20x18 Background Tiles

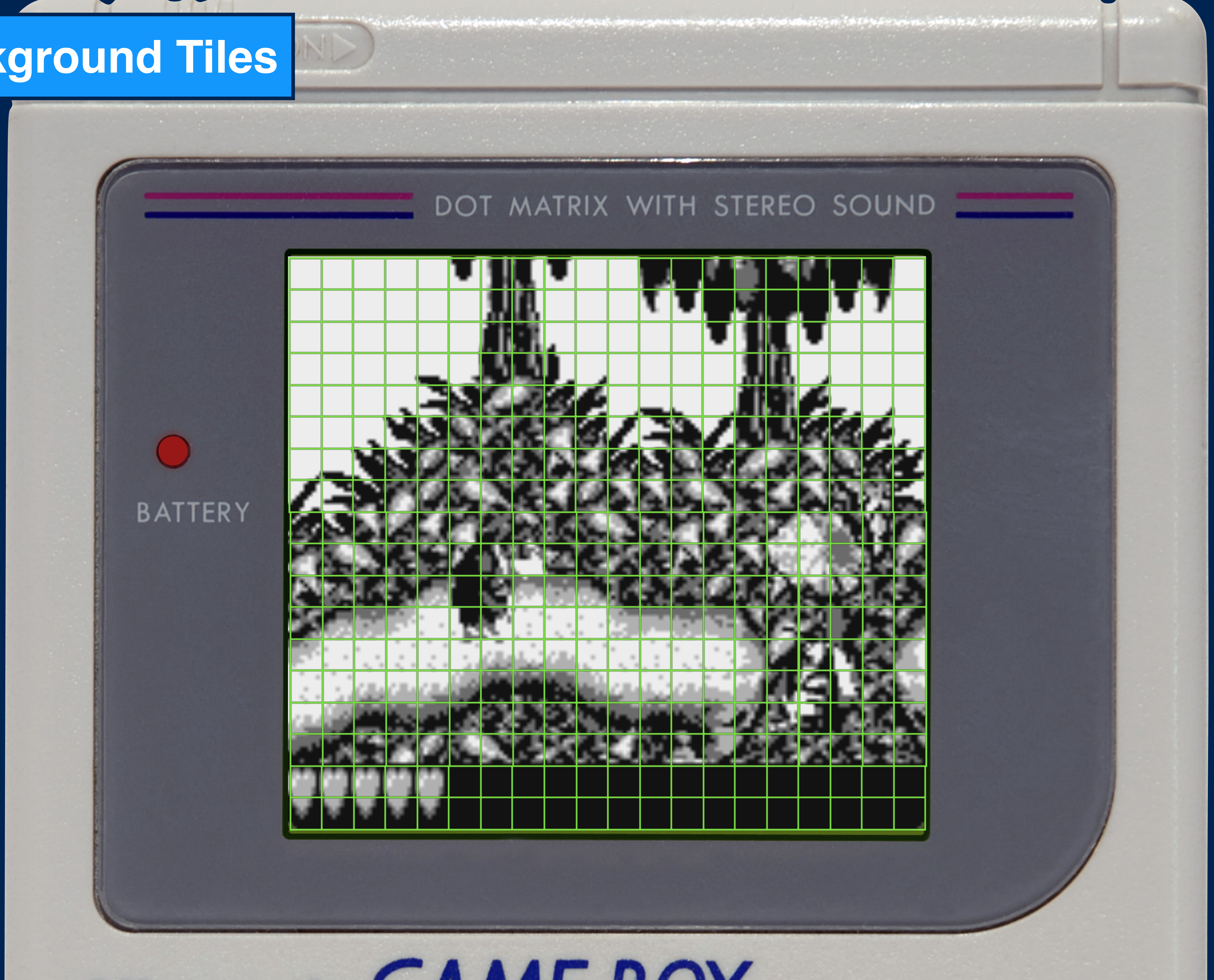
20x18 Background Tiles

20x18 Background Tiles



20x18 Background Tiles

20x18 Background Tiles

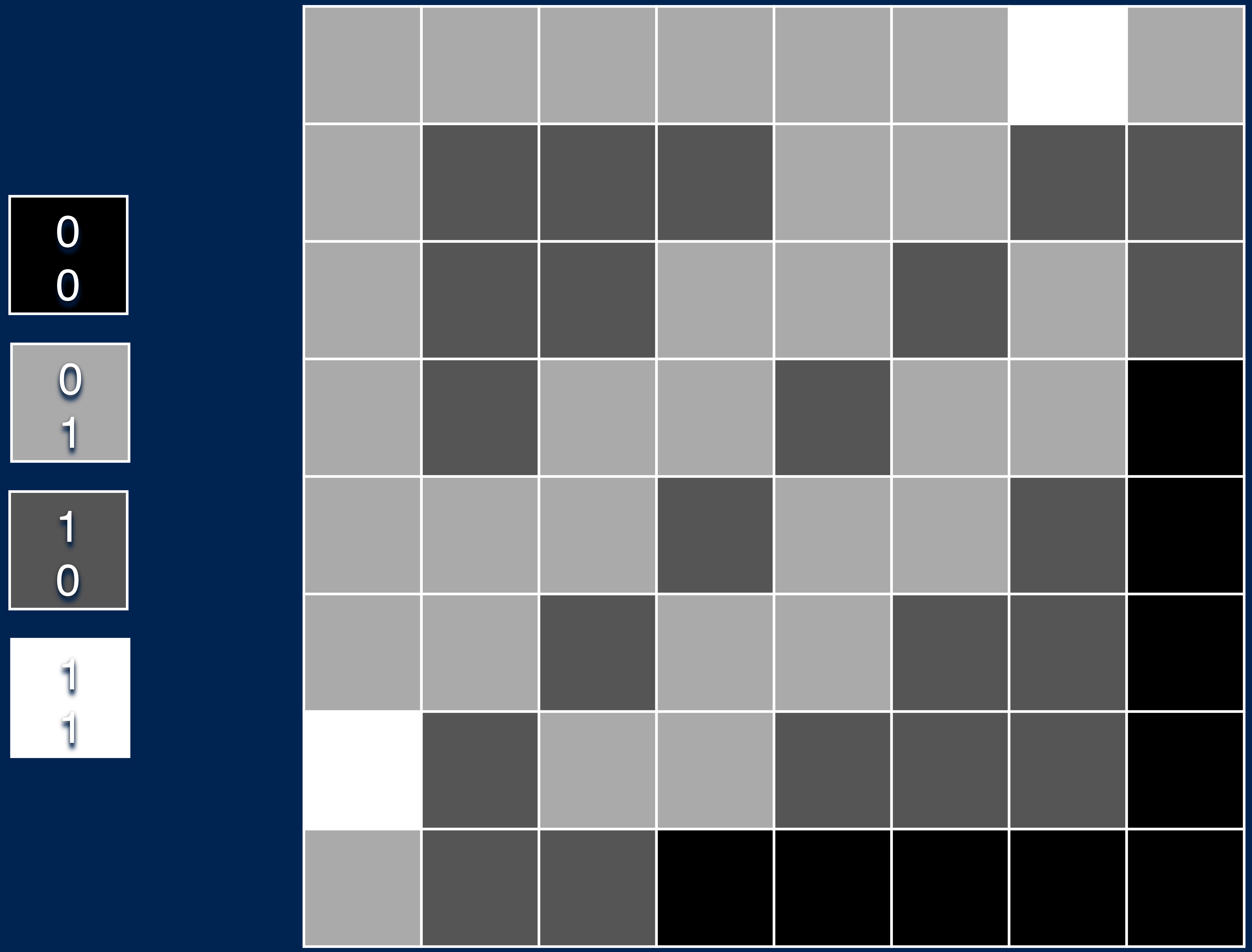


20x18 Background Tiles

20x18 Background Tiles

20x18 Background Tiles

8x8 Pixel Tiles



8x8 Pixel Tiles

00	01	01	01	01	01	01	11	01
01	01	10	10	10	01	01	10	10
10	01	10	10	01	01	10	01	10
11	01	10	01	01	10	01	01	00
	01	01	01	10	01	01	10	00
	01	01	10	01	01	10	10	00
	11	10	01	01	10	10	10	00
	01	10	10	00	00	00	00	00

8x8 Pixel Tiles

02

- 0
0
- 0
1
- 1
0
- 1
1

0 1	0 1	0 1	0 1	0 1	0 1	1 1	0 1
0 1	1 0	1 0	1 0	0 1	0 1	1 0	1 0
0 1	1 0	1 0	0 1	0 1	1 0	0 1	1 0
0 1	1 0	0 1	0 1	1 0	0 1	0 1	0 0
0 1	0 1	0 1	1 0	0 1	0 1	1 0	0 0
0 1	0 1	1 0	0 1	0 1	1 0	1 0	0 0
1 1	1 0	0 1	0 1	1 0	1 0	1 0	0 0
0 1	1 0	1 0	0 0	0 0	0 0	0 0	0 0

8x8 Pixel Tiles

0 0
0 1
1 0
1 1

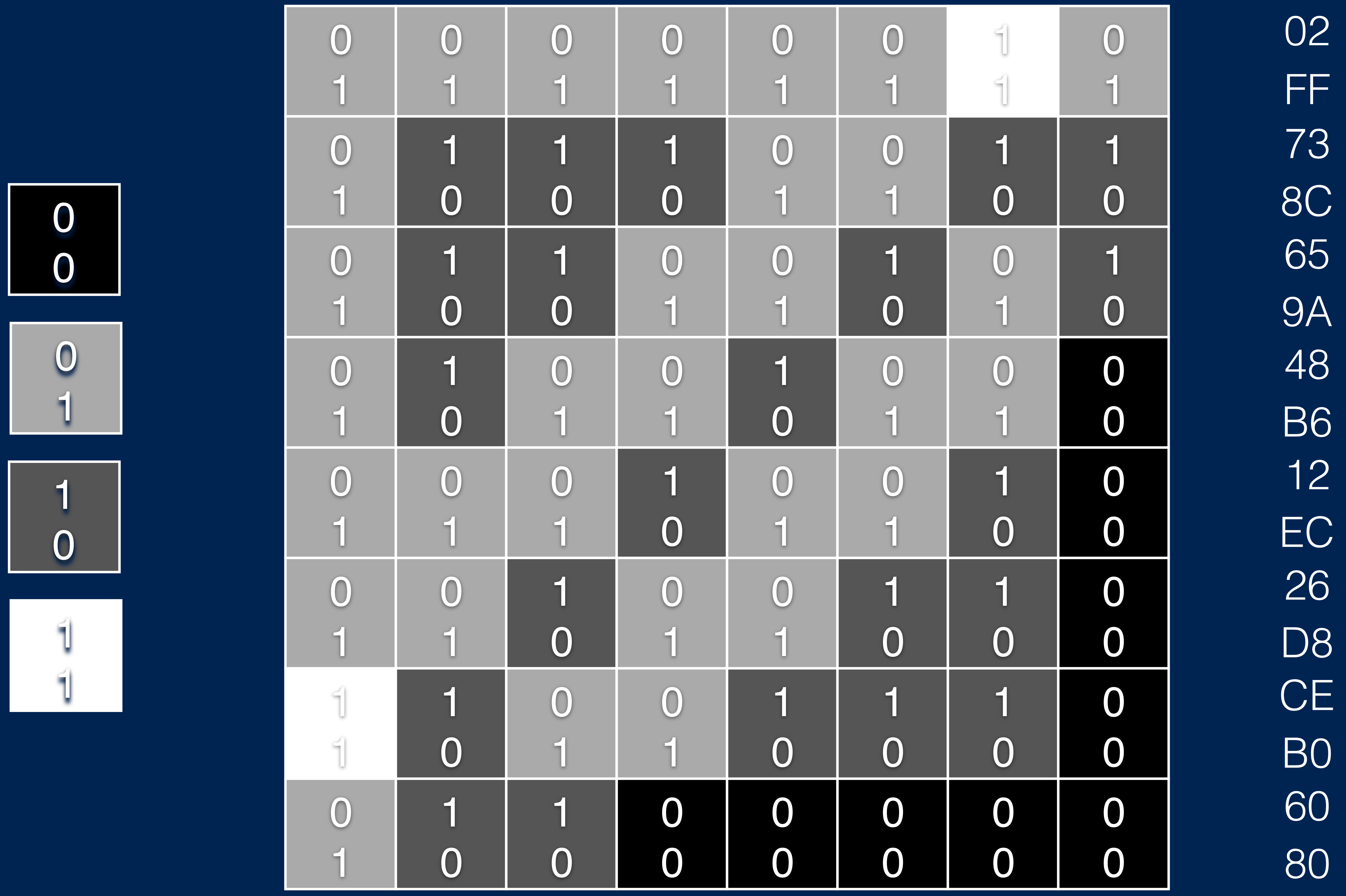
0 1	0 1	0 1	0 1	0 1	0 1	1 1	0 1
0 1	1 0	1 0	1 0	0 1	0 1	1 0	1 0
0 1	1 0	1 0	0 1	0 1	1 0	0 1	1 0
0 1	1 0	0 1	0 1	1 0	0 1	0 1	0 0
0 1	0 1	0 1	1 0	0 1	0 1	1 0	0 0
0 1	0 1	1 0	0 1	0 1	1 0	1 0	0 0
1 1	1 0	0 1	0 1	1 0	1 0	1 0	0 0
0 1	1 0	1 0	0 0	0 0	0 0	0 0	0 0

02
FF

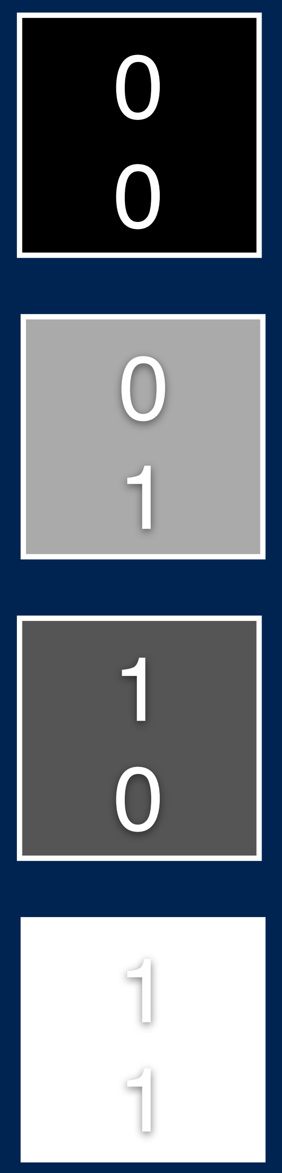
8x8 Pixel Tiles

0 1	0 1	0 1	0 1	0 1	0 1	1 1	0 1	02 FF
0 1	1 0	1 0	1 0	0 1	0 1	1 0	1 0	73 8C
0 1	1 0	1 0	0 1	0 1	1 0	0 1	1 0	65 9A
0 1	1 0	0 1	0 1	1 0	0 1	0 1	0 0	48 B6
0 1	0 1	0 1	1 0	0 1	0 1	1 0	0 0	12 EC
0 1	0 1	1 0	0 1	0 1	1 0	1 0	0 0	26 D8
1 1	1 0	0 1	0 1	1 0	1 0	1 0	0 0	CE B0
0 1	1 0	1 0	0 0	0 0	0 0	0 0	0 0	60 80

8x8 Pixel Tiles



16 bytes per tile


0 0
0 1
1 0
1 1

Background Palettes

FF47	BGP	BG Palette (R/W)
6-7		Color for 11
4-5		Color for 10
2-3		Color for 01
0-1		Color for 00

00	01	10	11

0
0

0
1

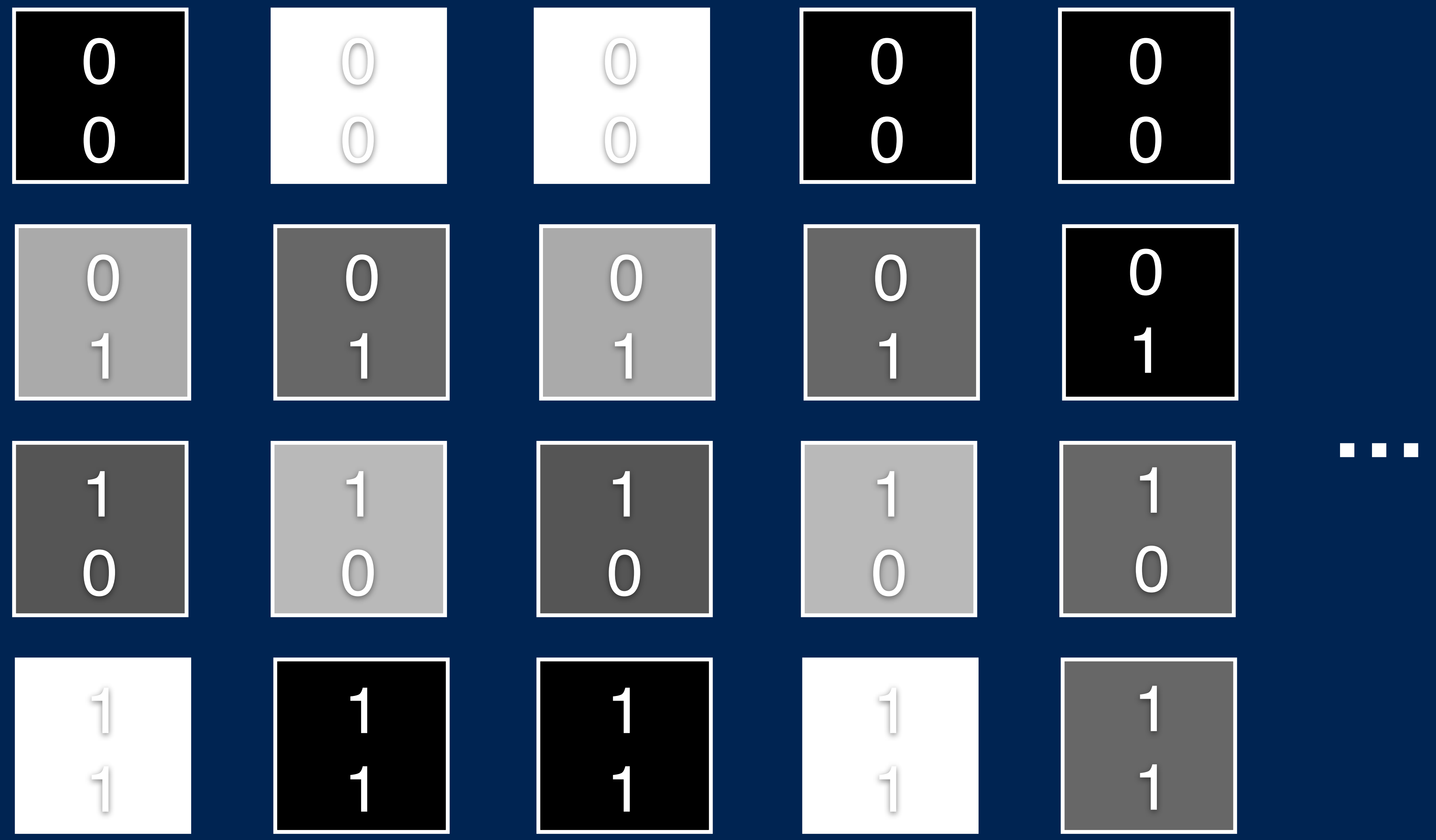
1
0

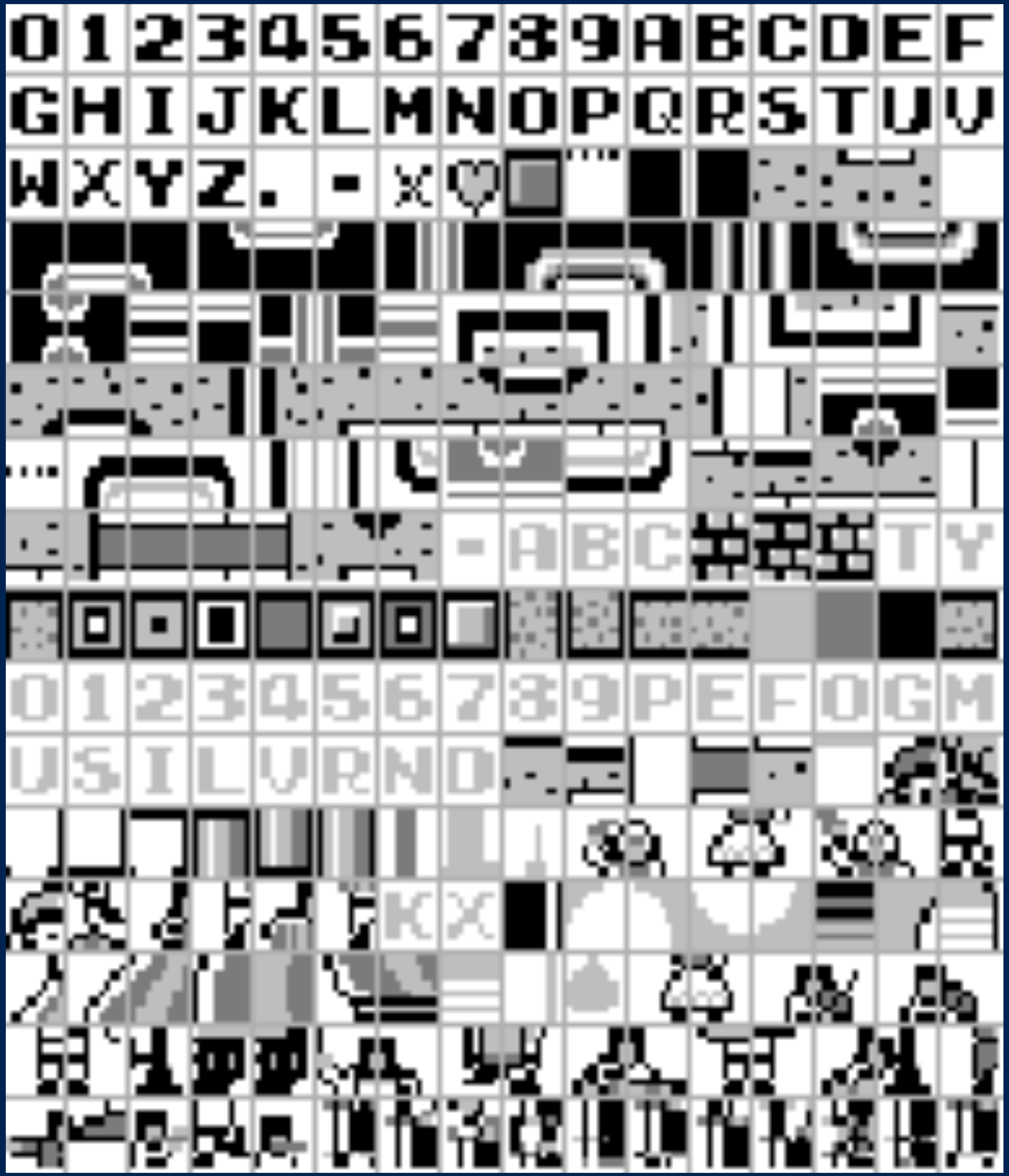
1
1

Background Palettes

FF47	BGP	BG Palette (R/W)
6-7		Color for 11
4-5		Color for 10
2-3		Color for 01
0-1		Color for 00

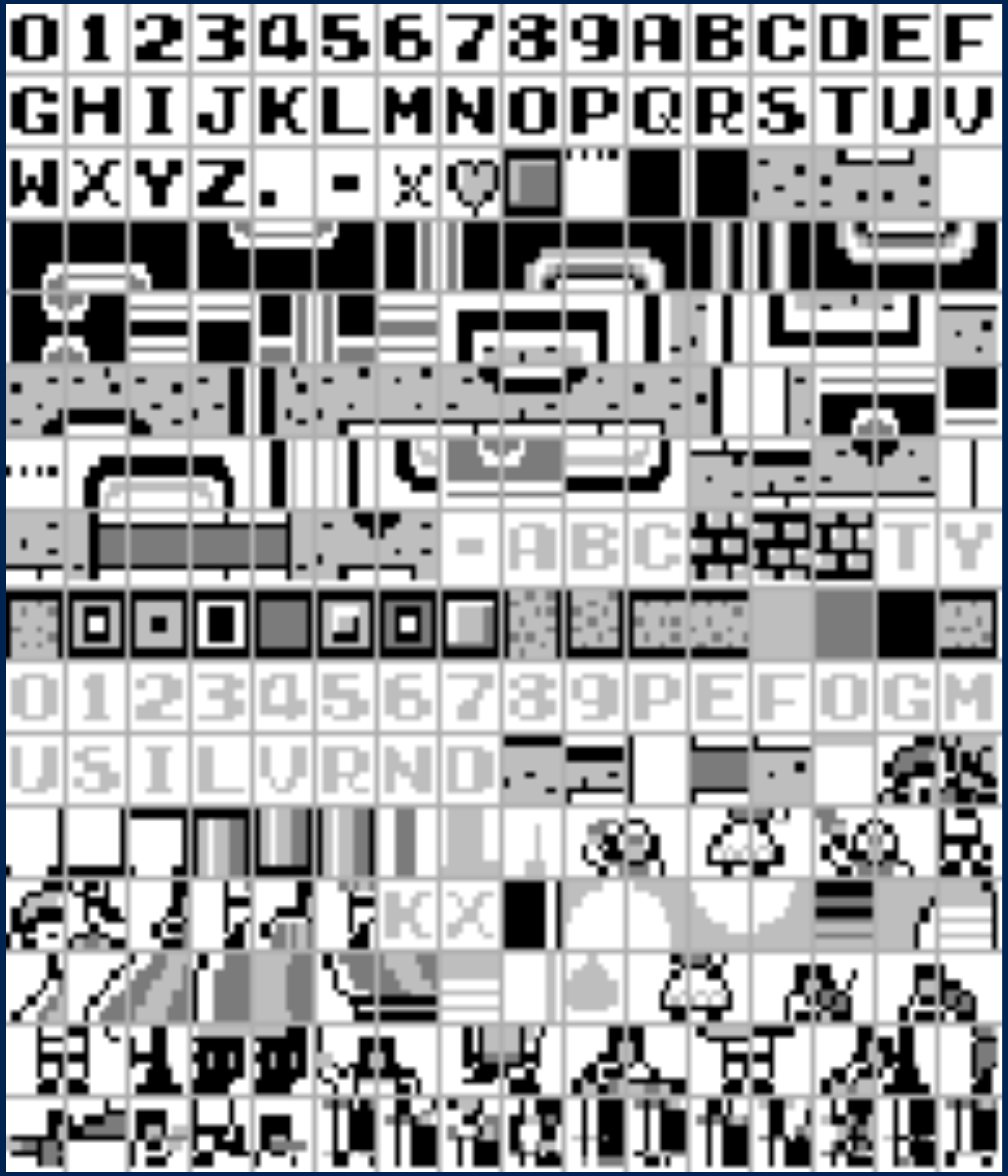
00	01	10	11





256 tiles

Background Tile Data



256 tiles

Background Tile Data

Tetris

256 tiles

Background Tile Data

Background Tile Data

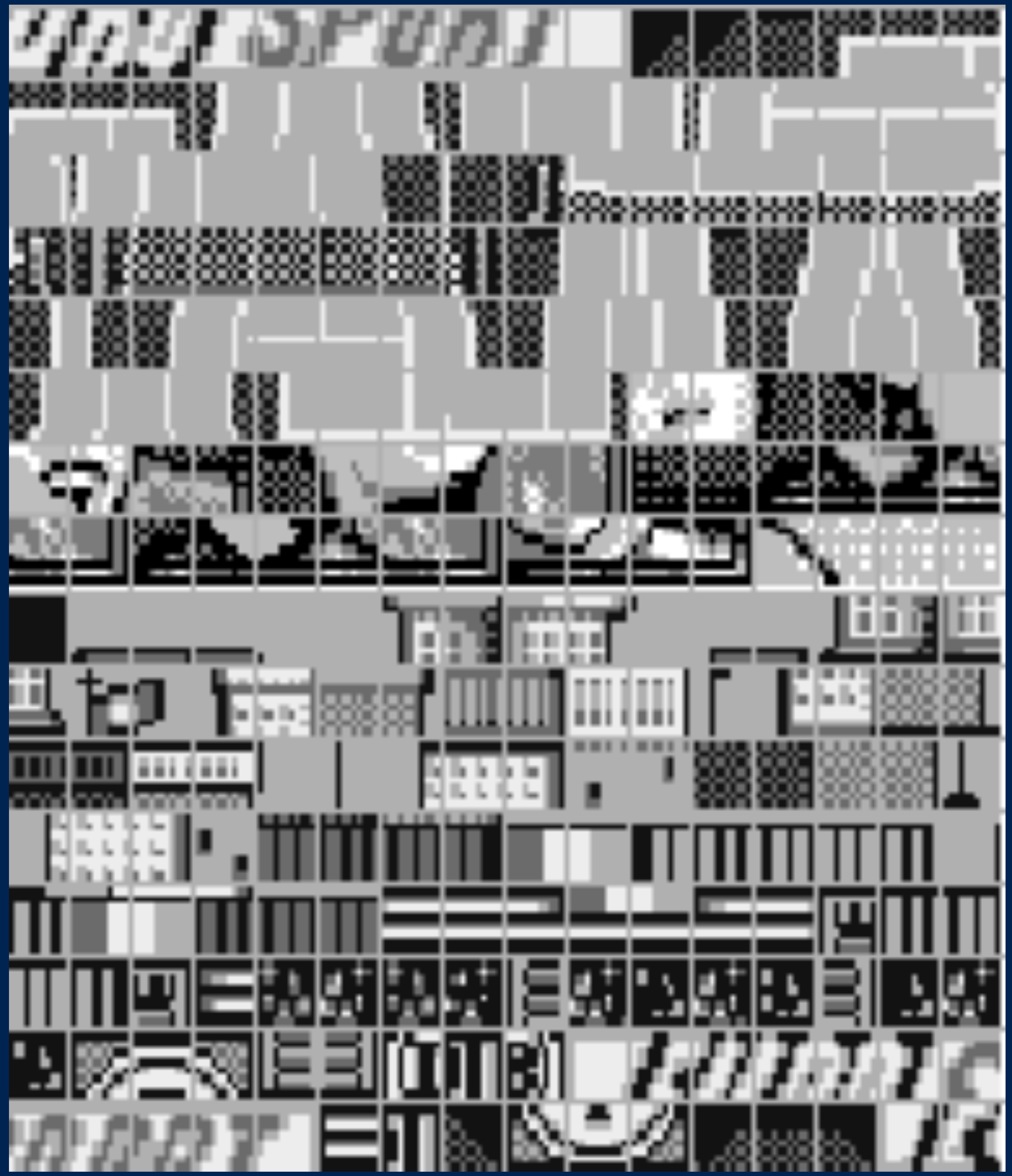
Zelda

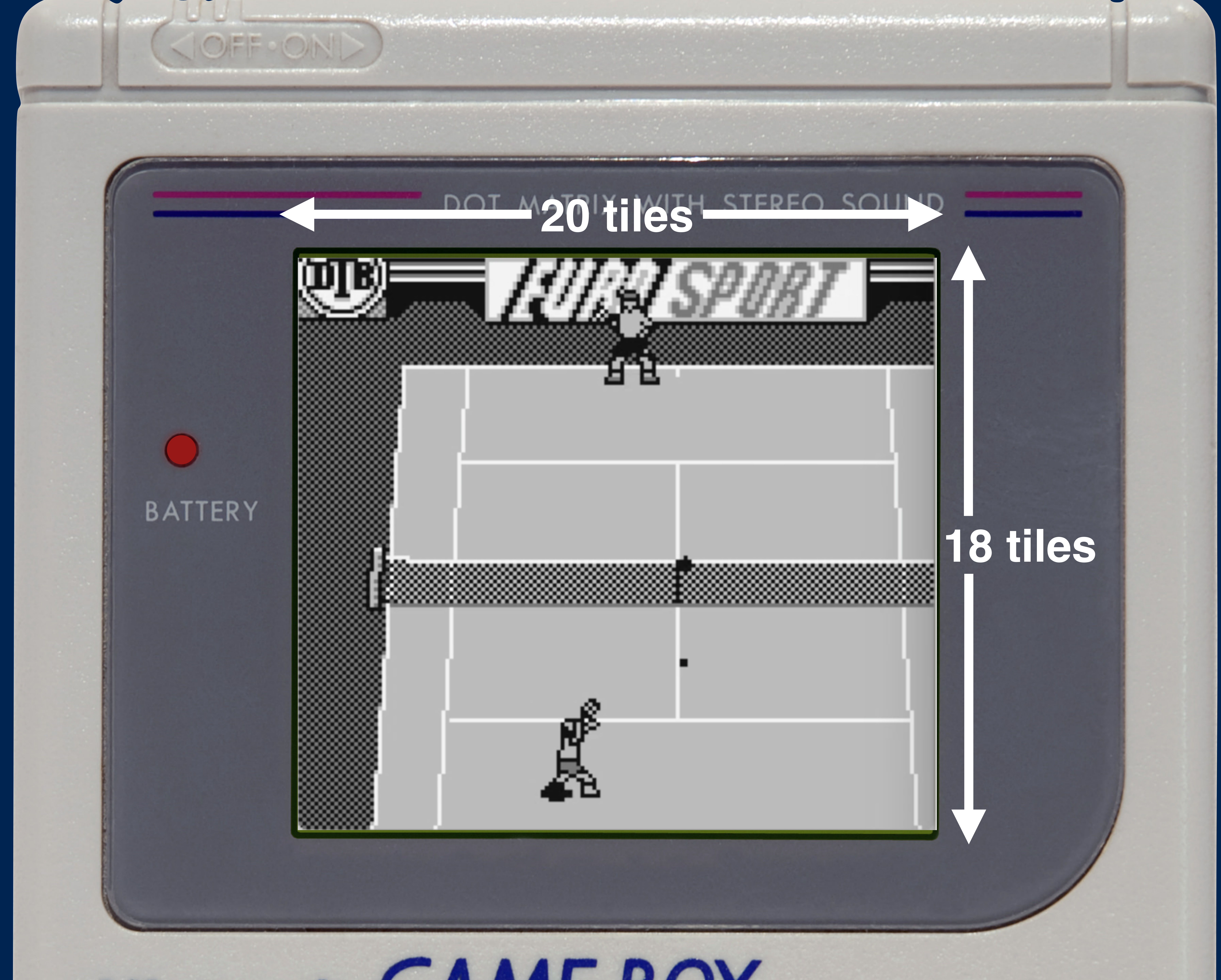
Background Tile Data

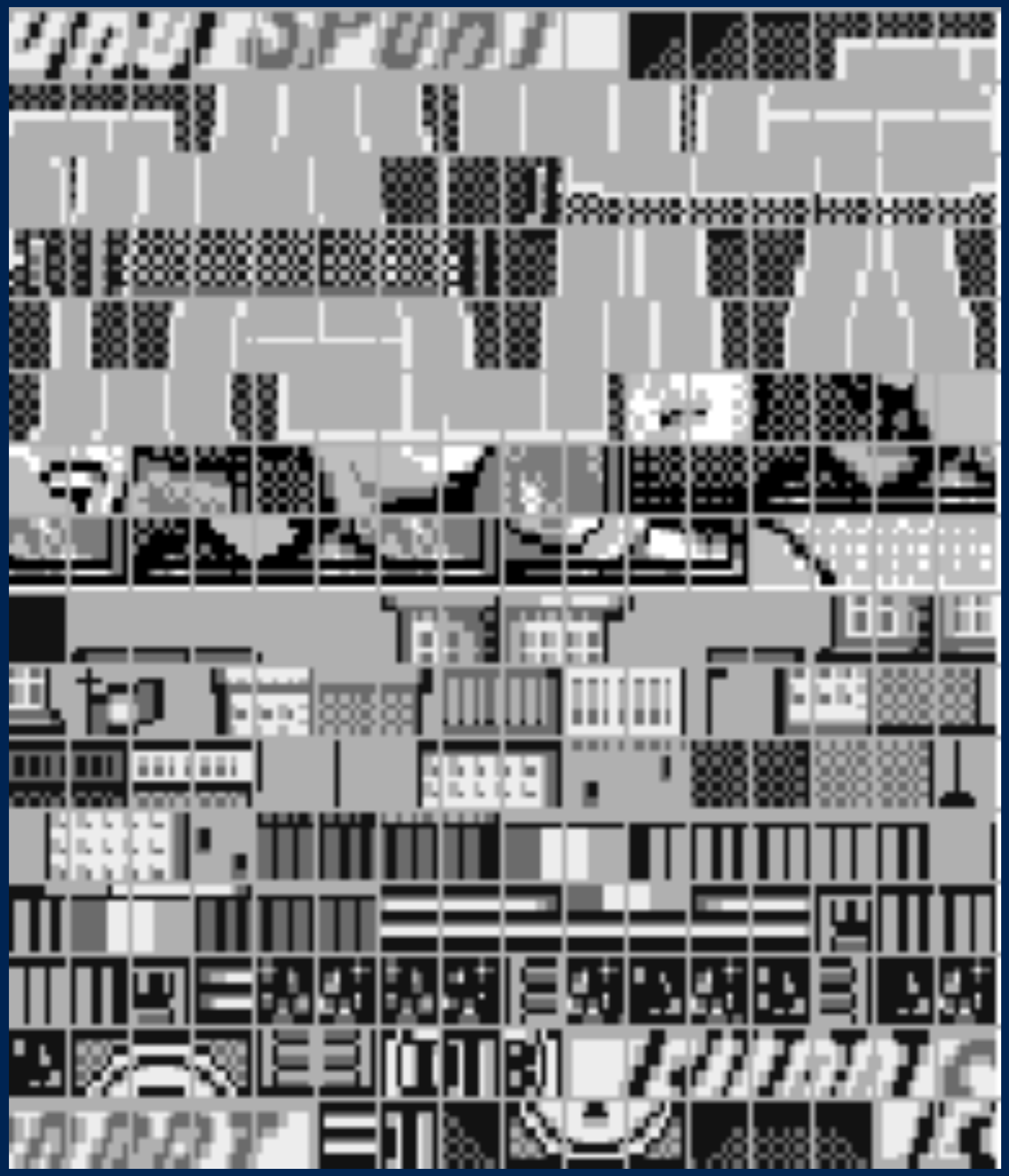
Background Tile Data

Super Mario Land

Background Tile Data


OFF · ON
DOT MATRIX WITH STEREO SOUND
BATTERY


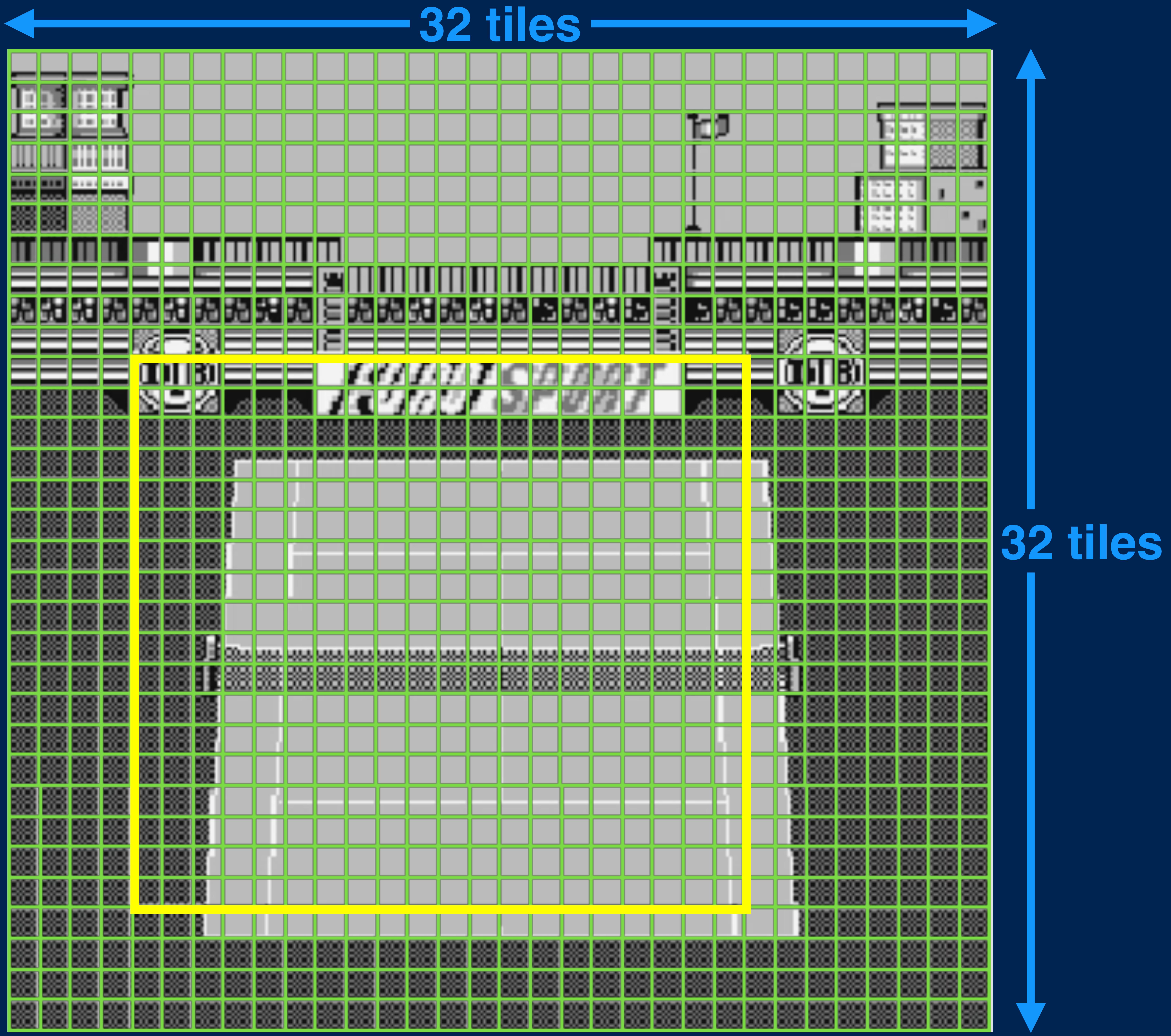




All Star Tennis

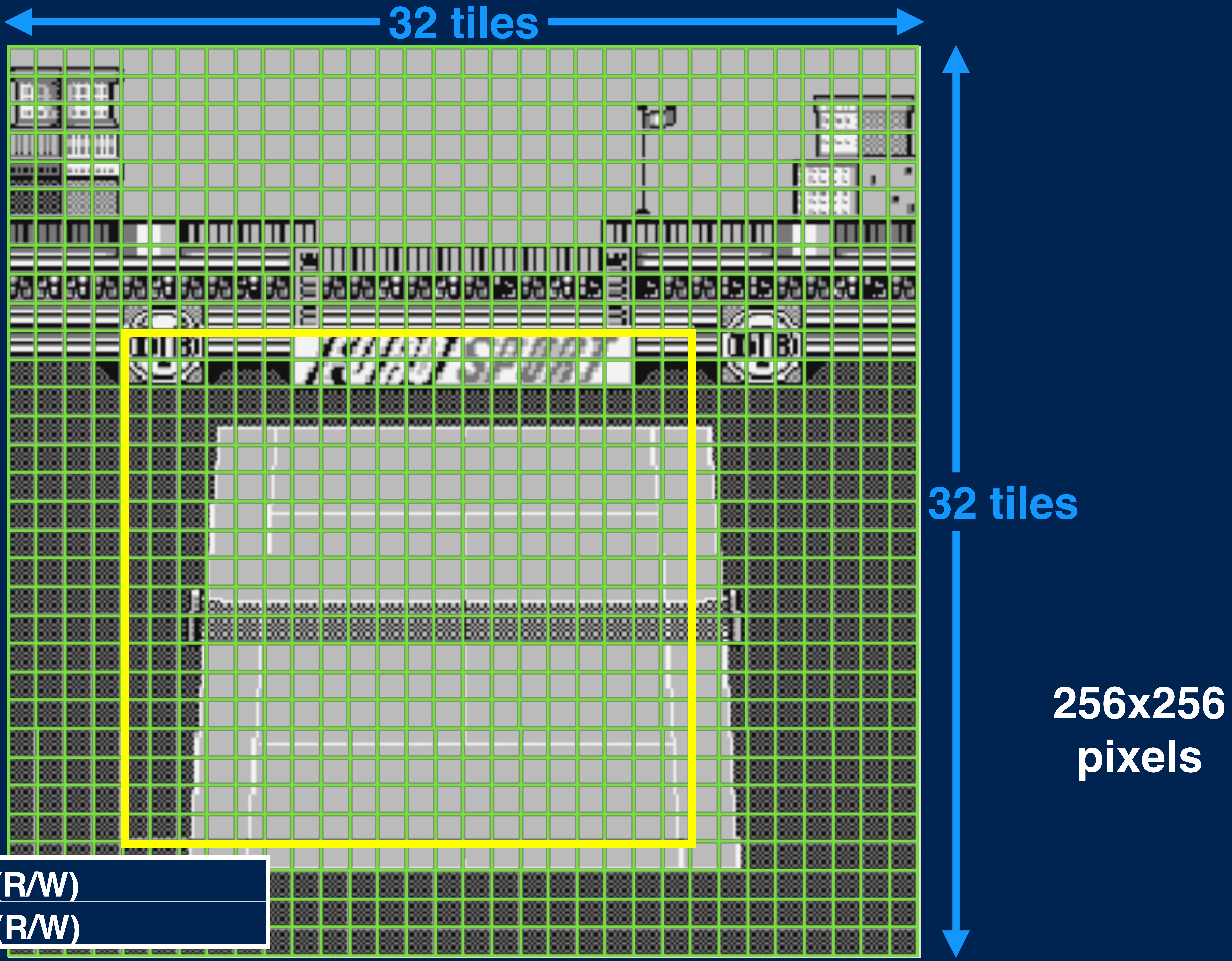

DTB
EURO SPORT

Background Map



Background Map

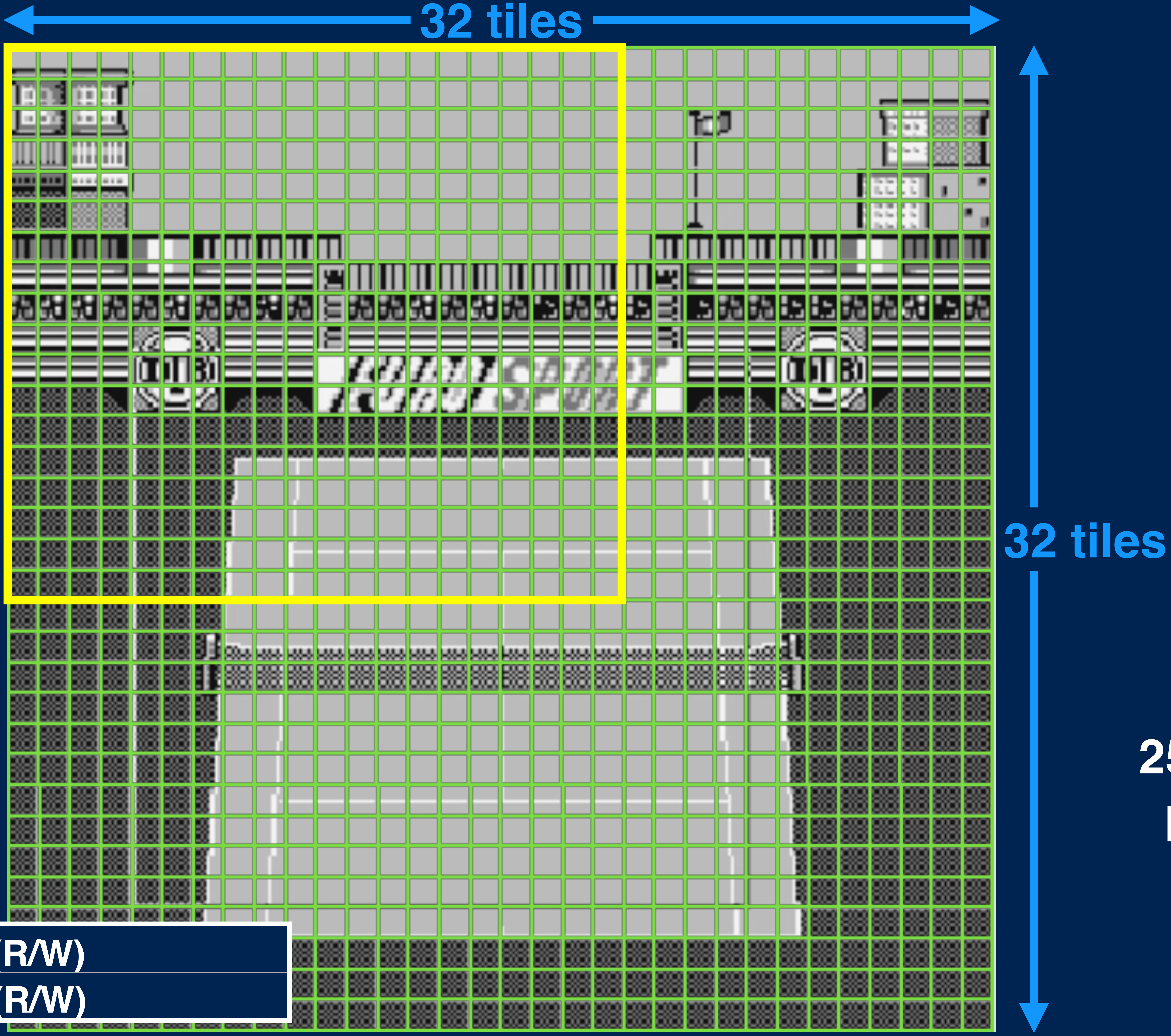
Scrolling



FF42	SCY	Scroll Y (R/W)
FF43	SCX	Scroll X (R/W)

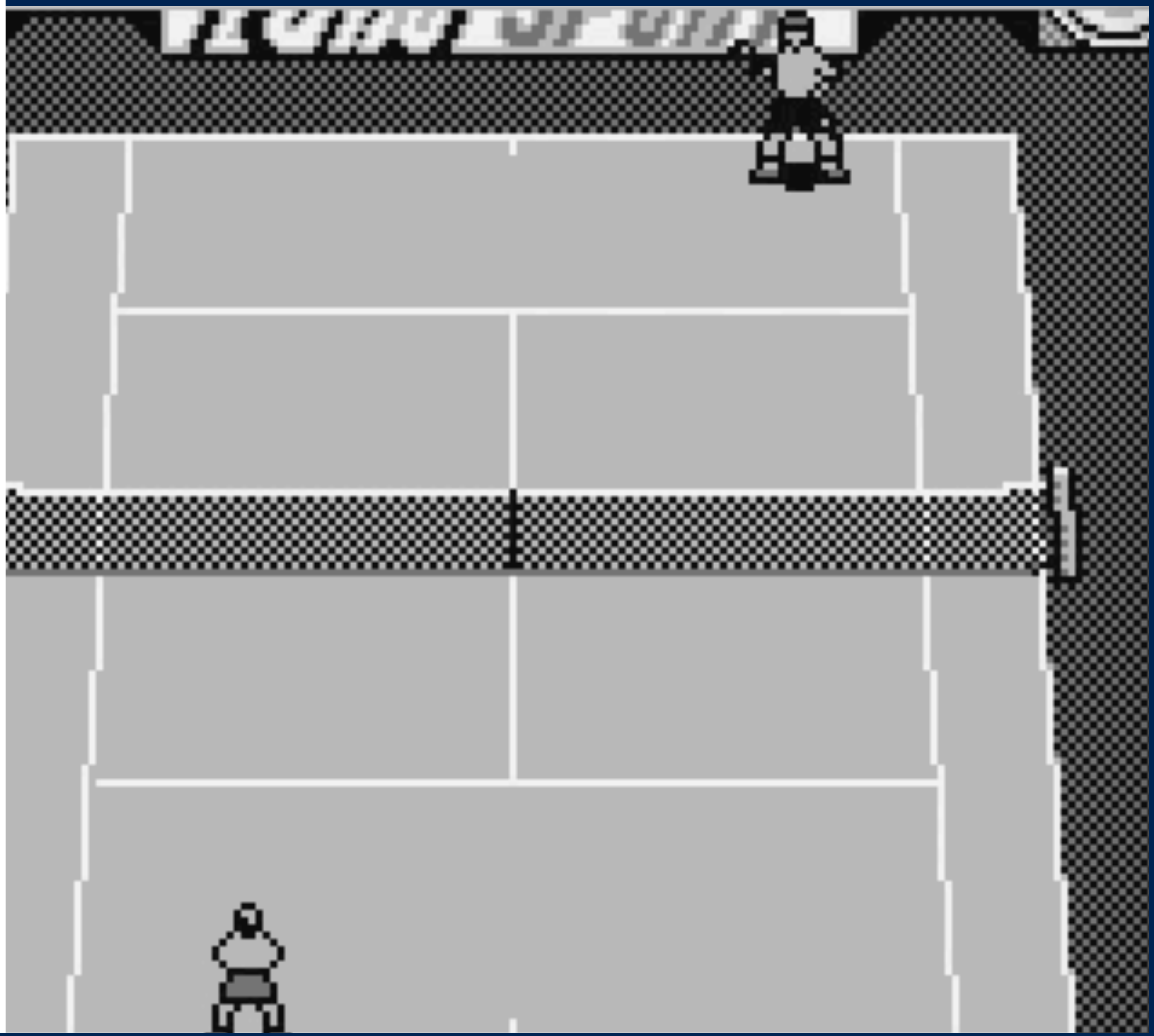
Background Map

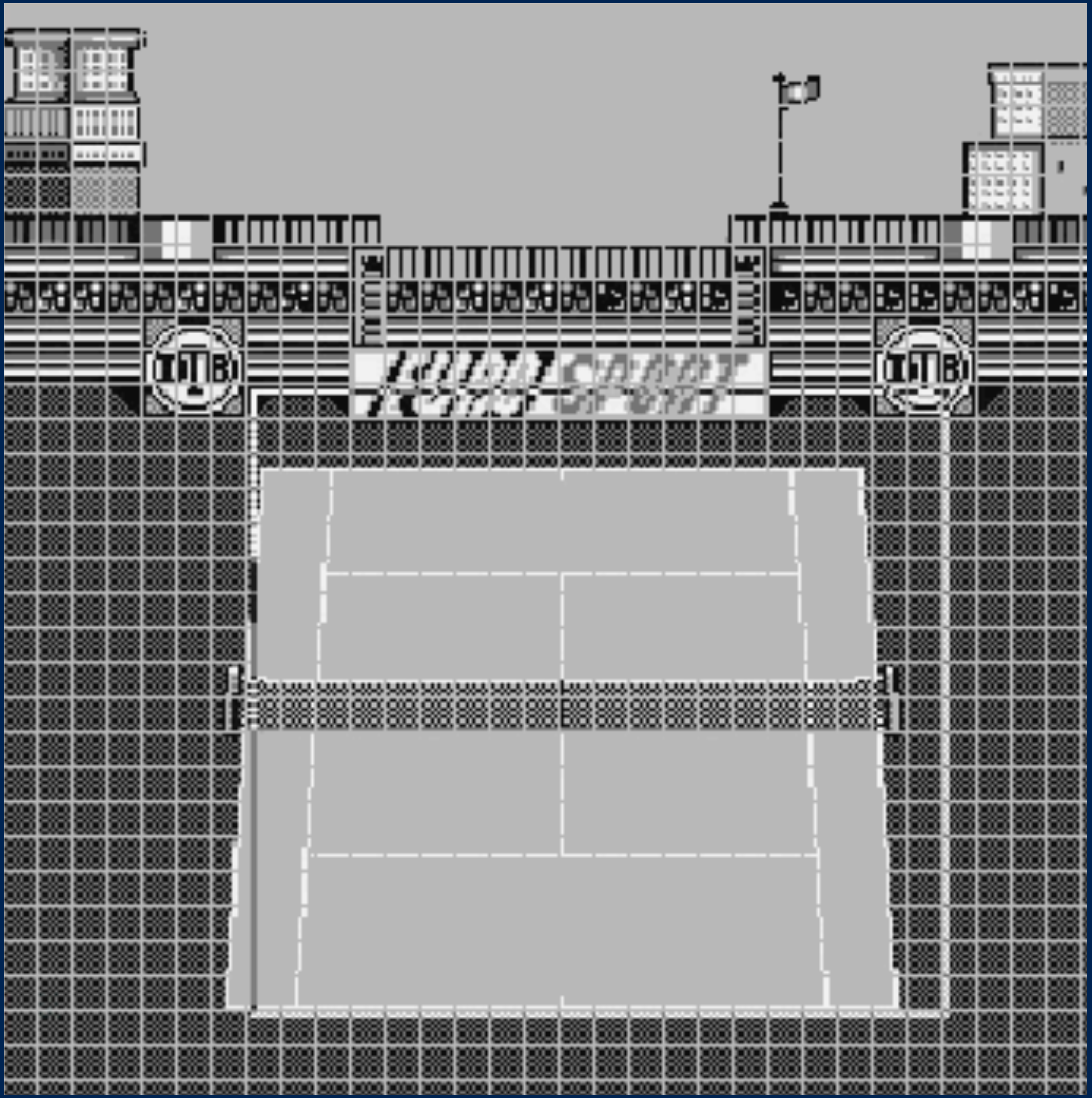
Scrolling

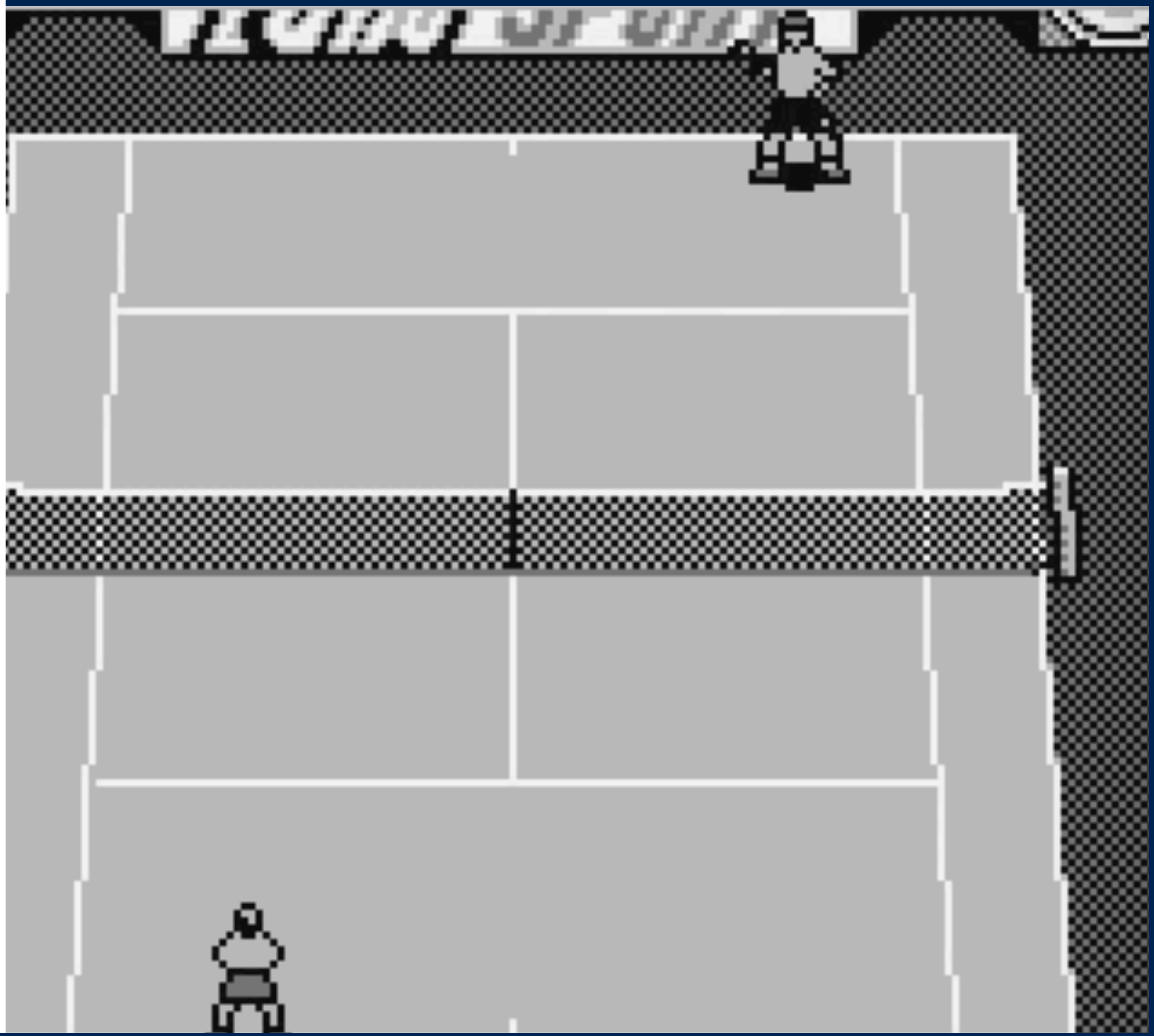


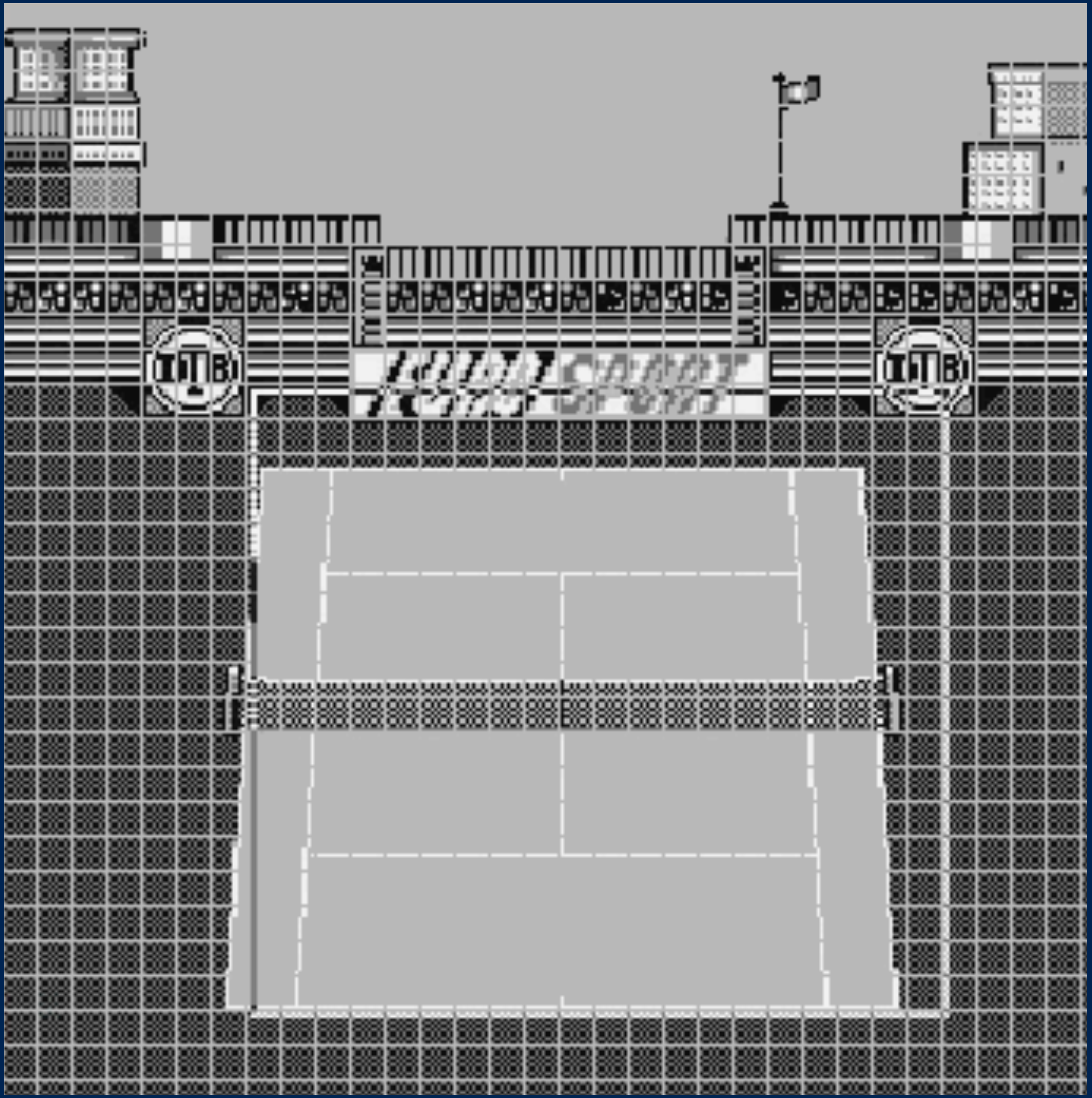
256x256 pixels

FF42	SCY	Scroll Y (R/W)
FF43	SCX	Scroll X (R/W)

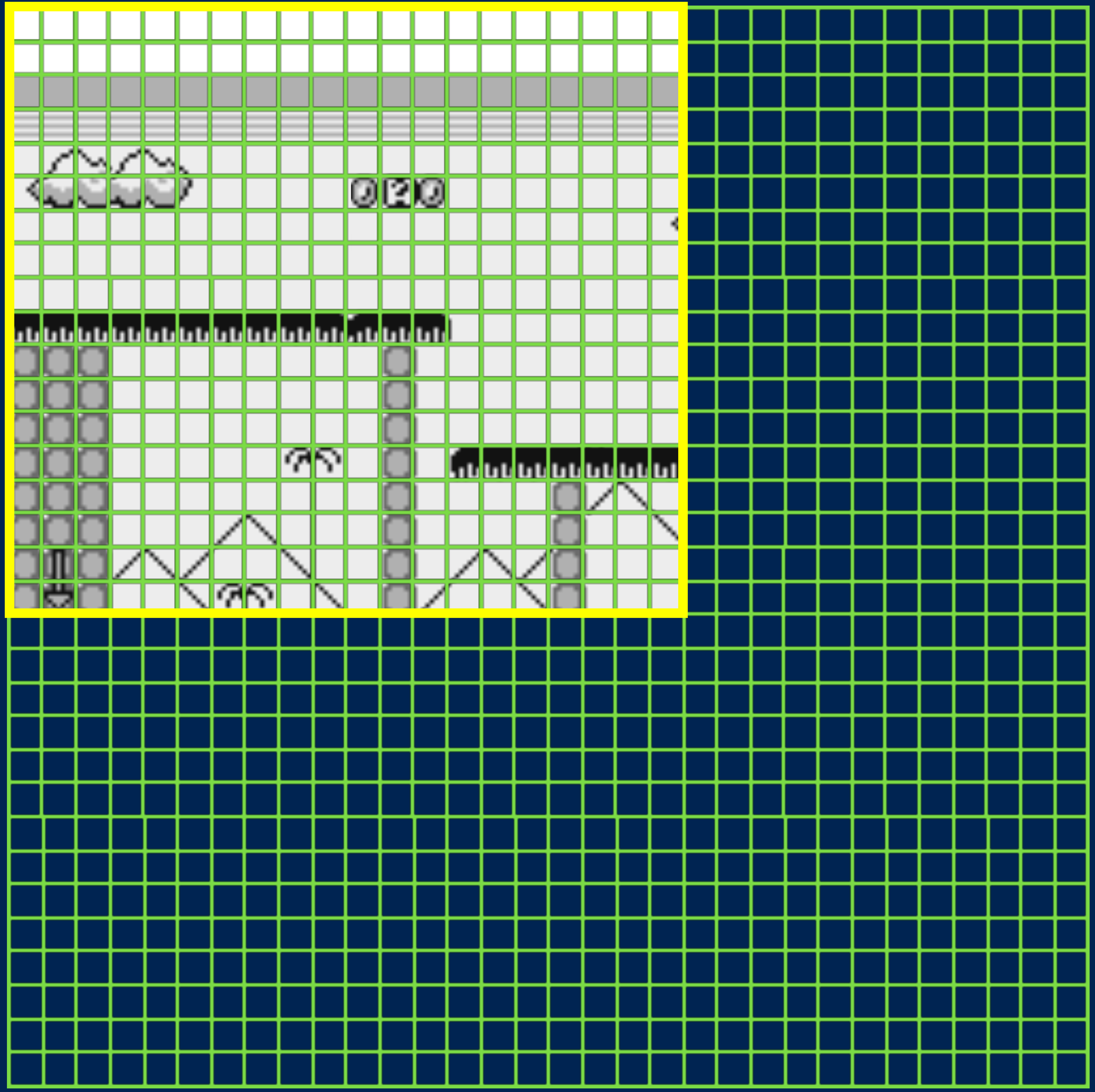


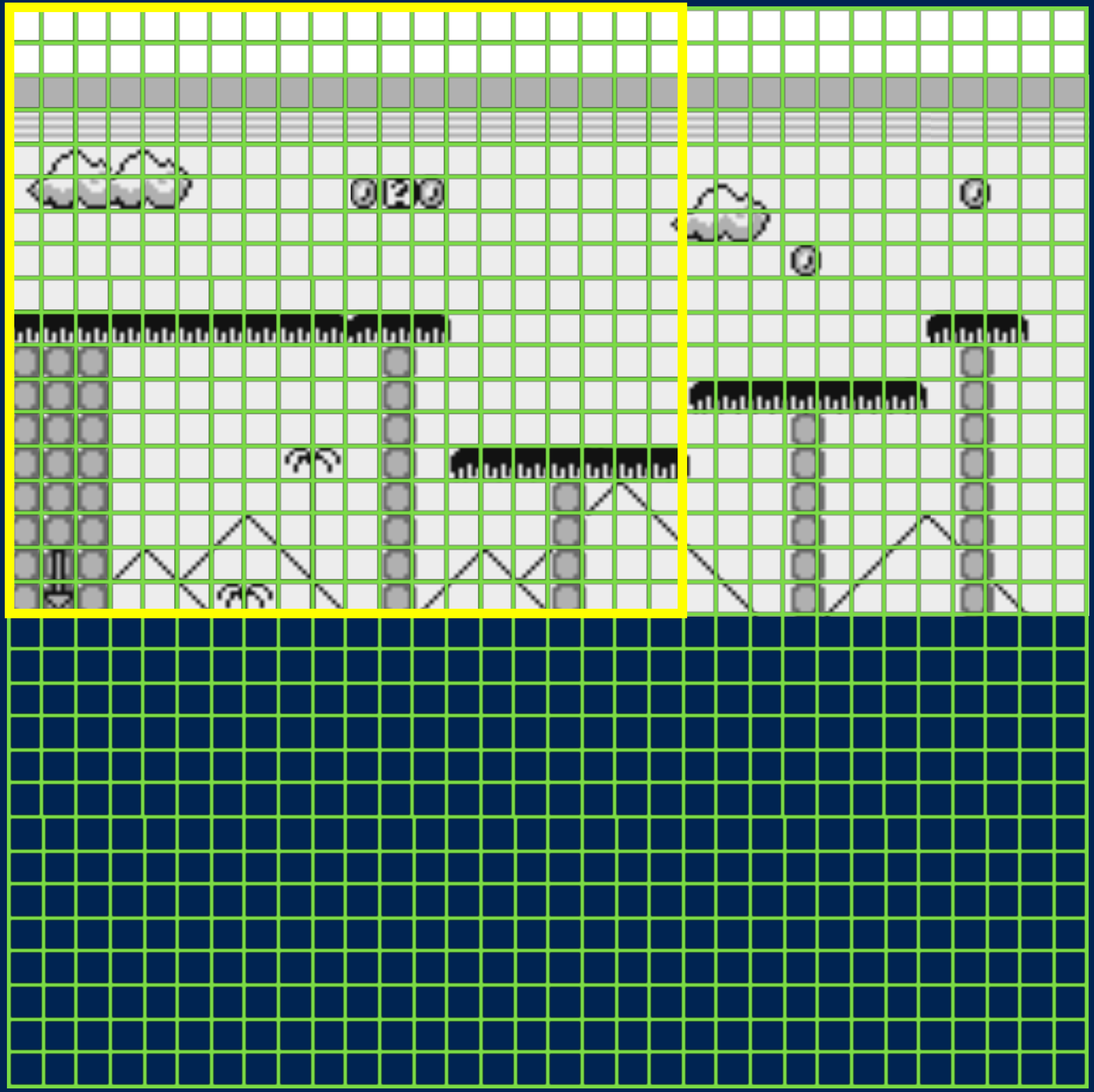


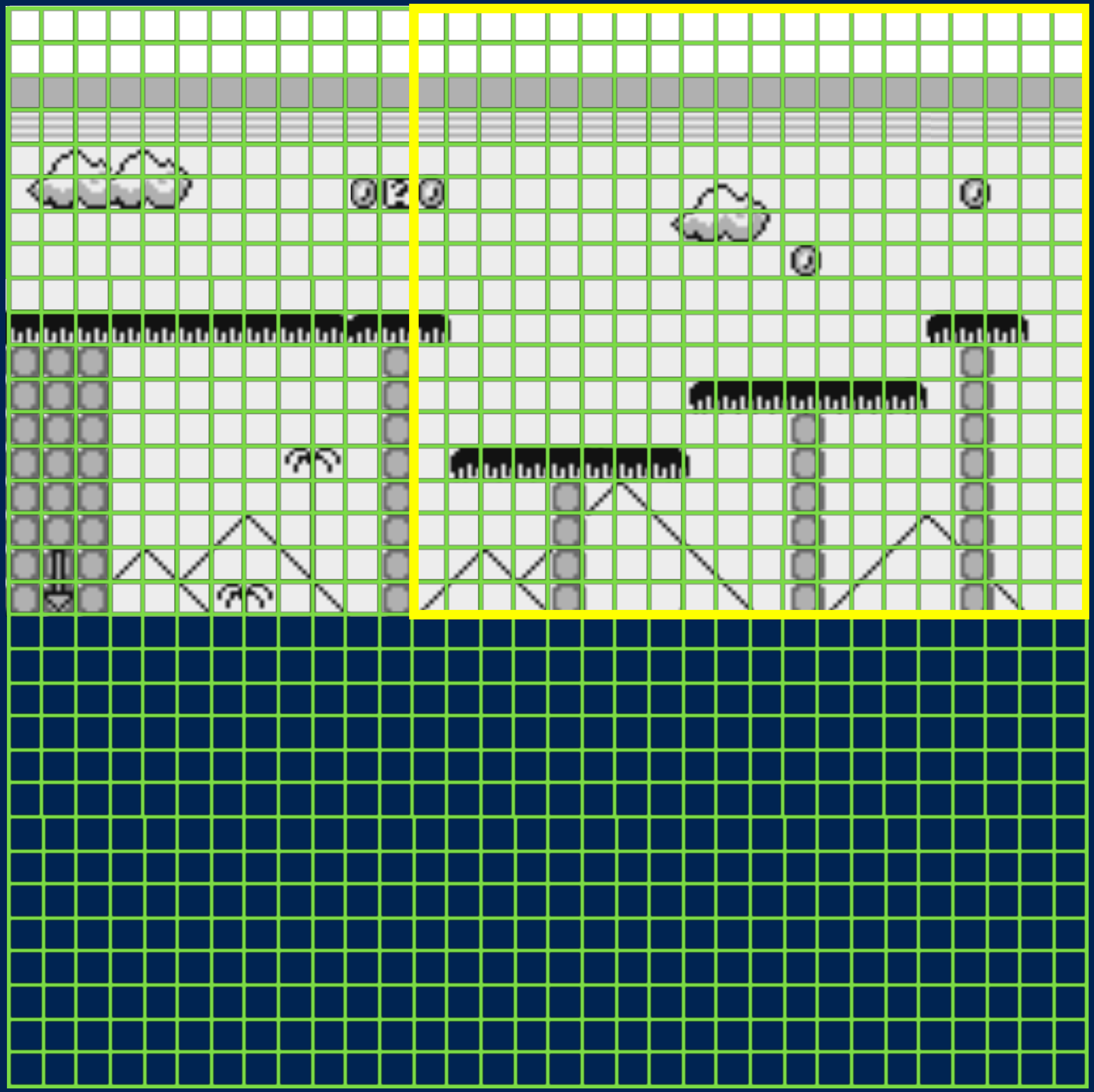


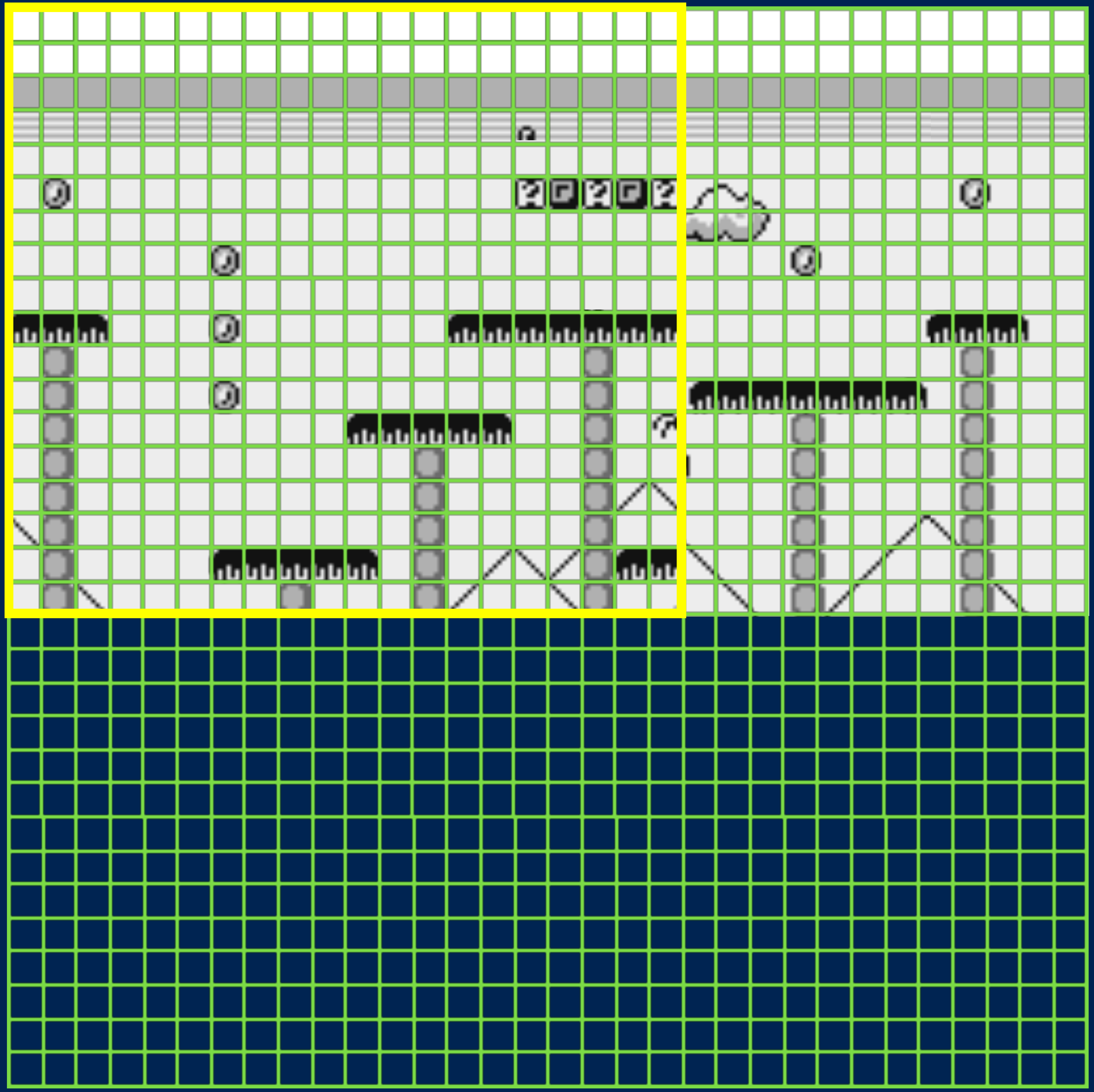

ITB
ITB

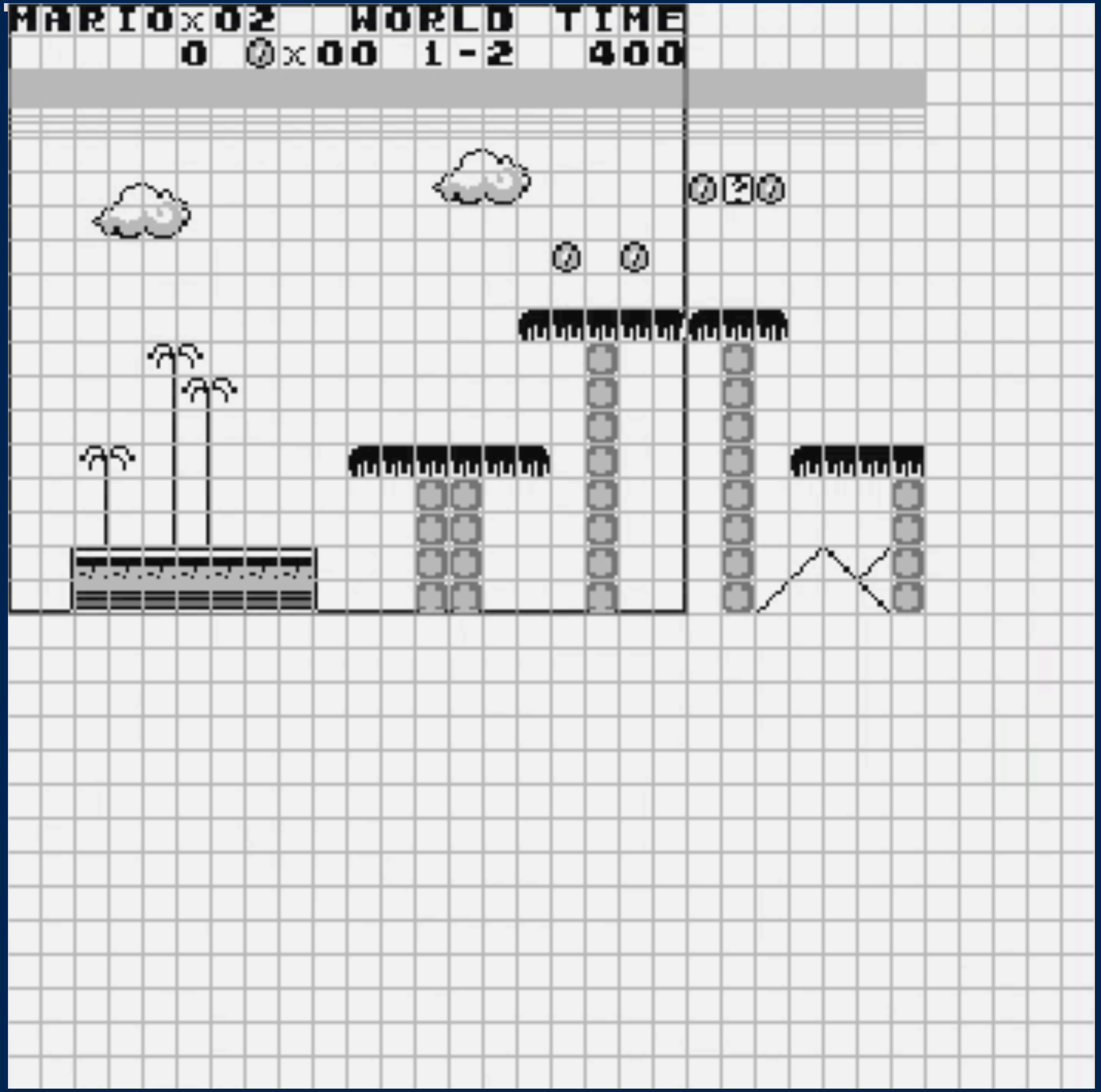

OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
MARIO×02 WORLD TIME
0 ×00 1-2 400

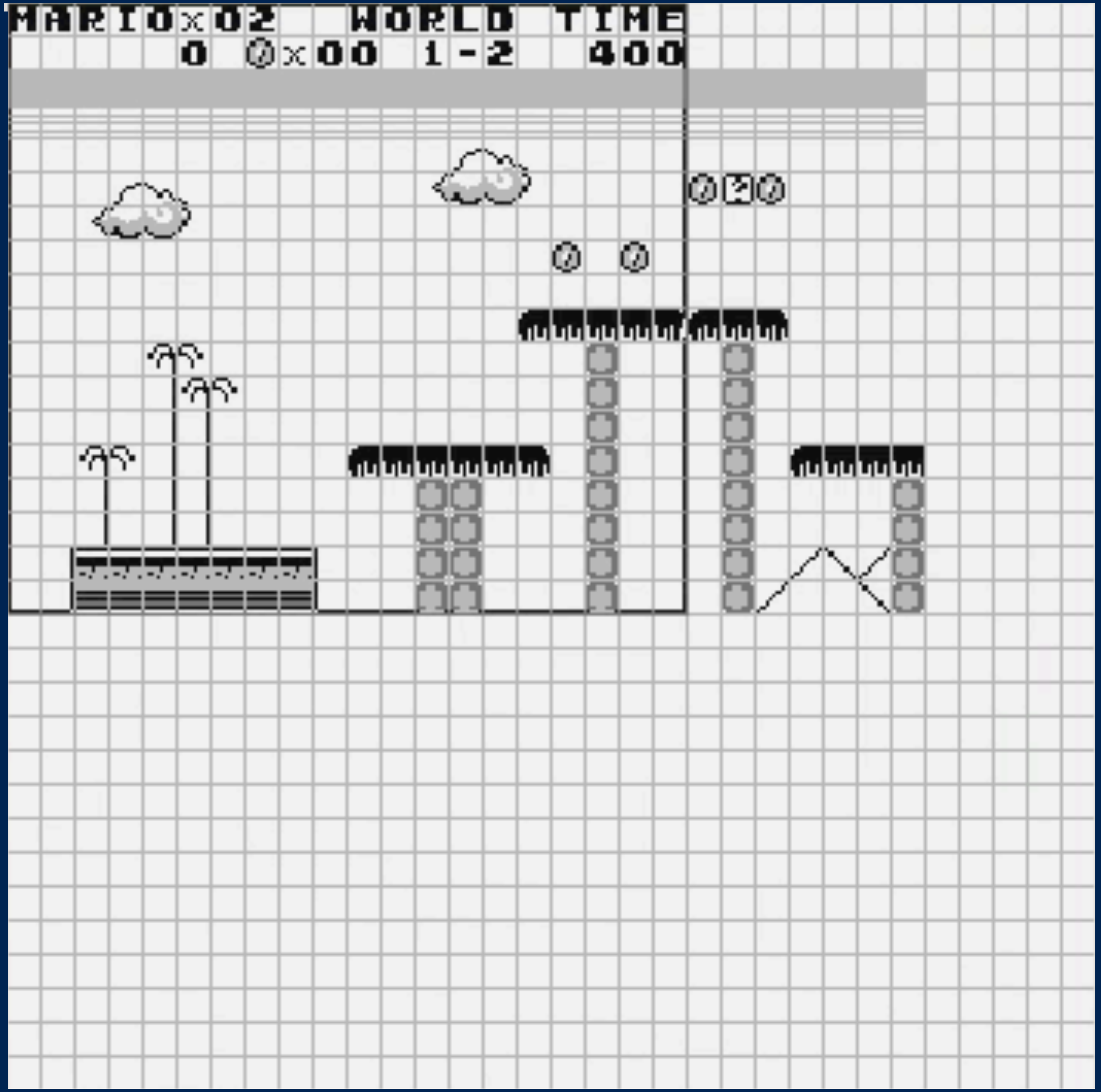


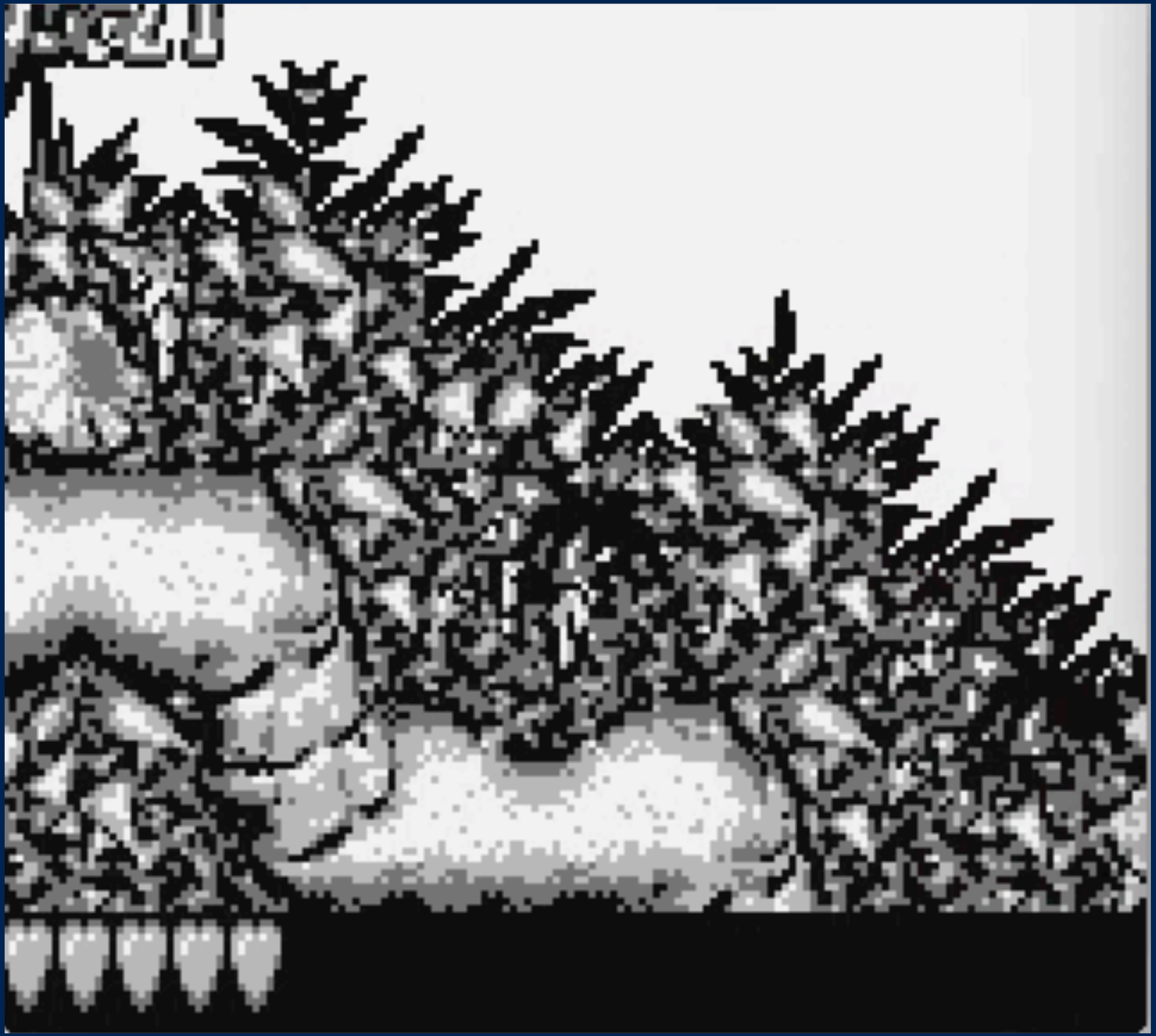


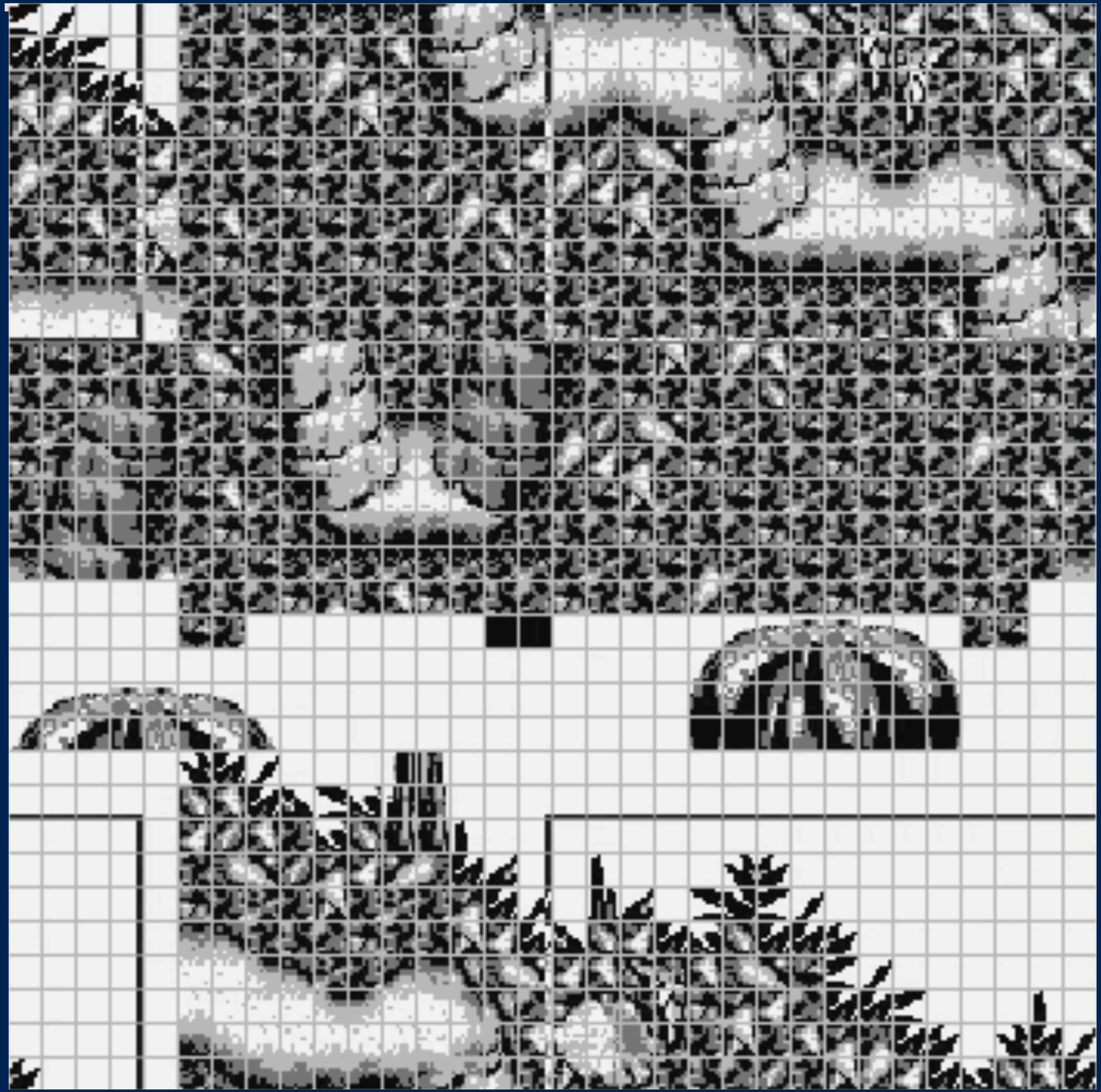


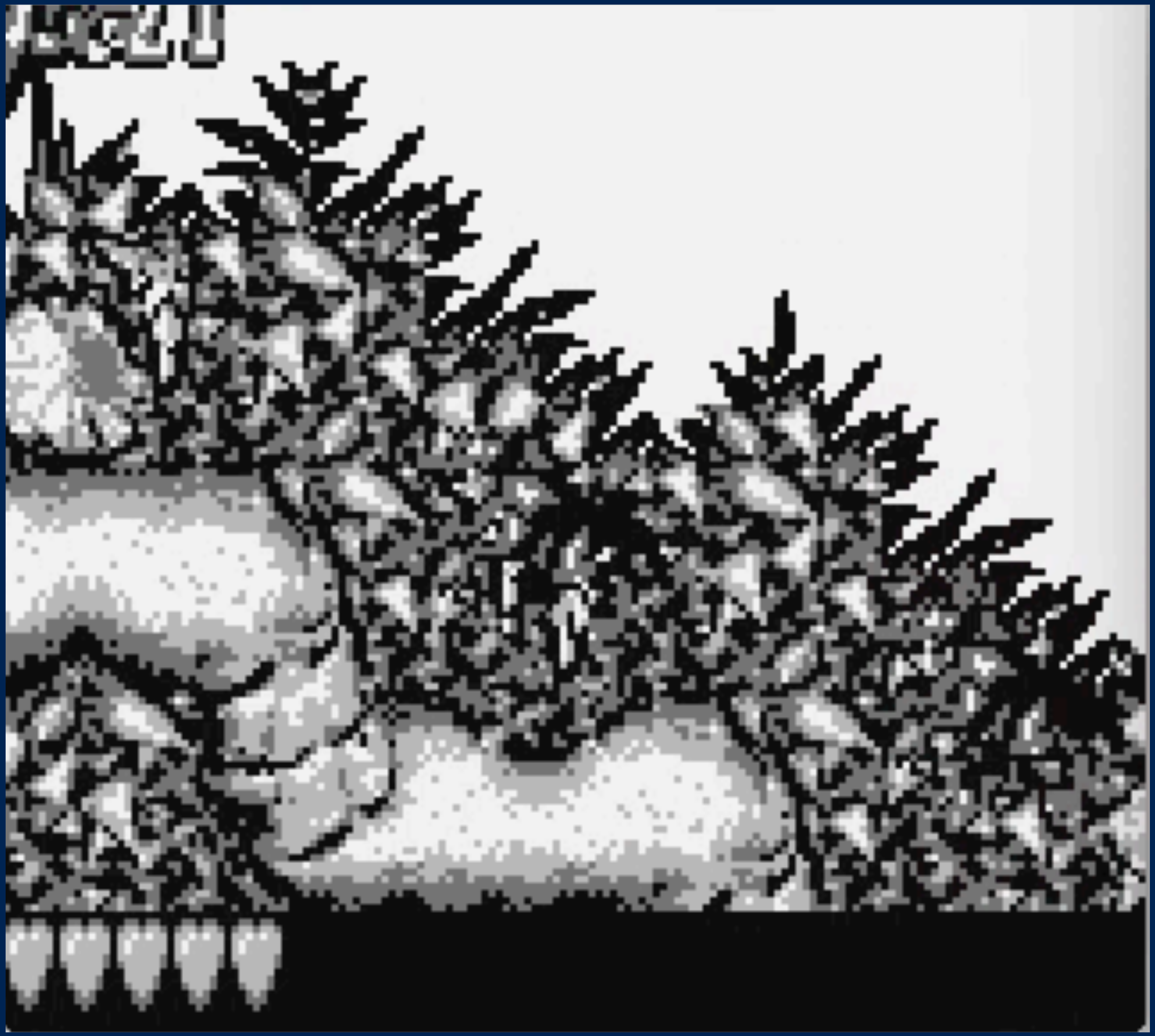


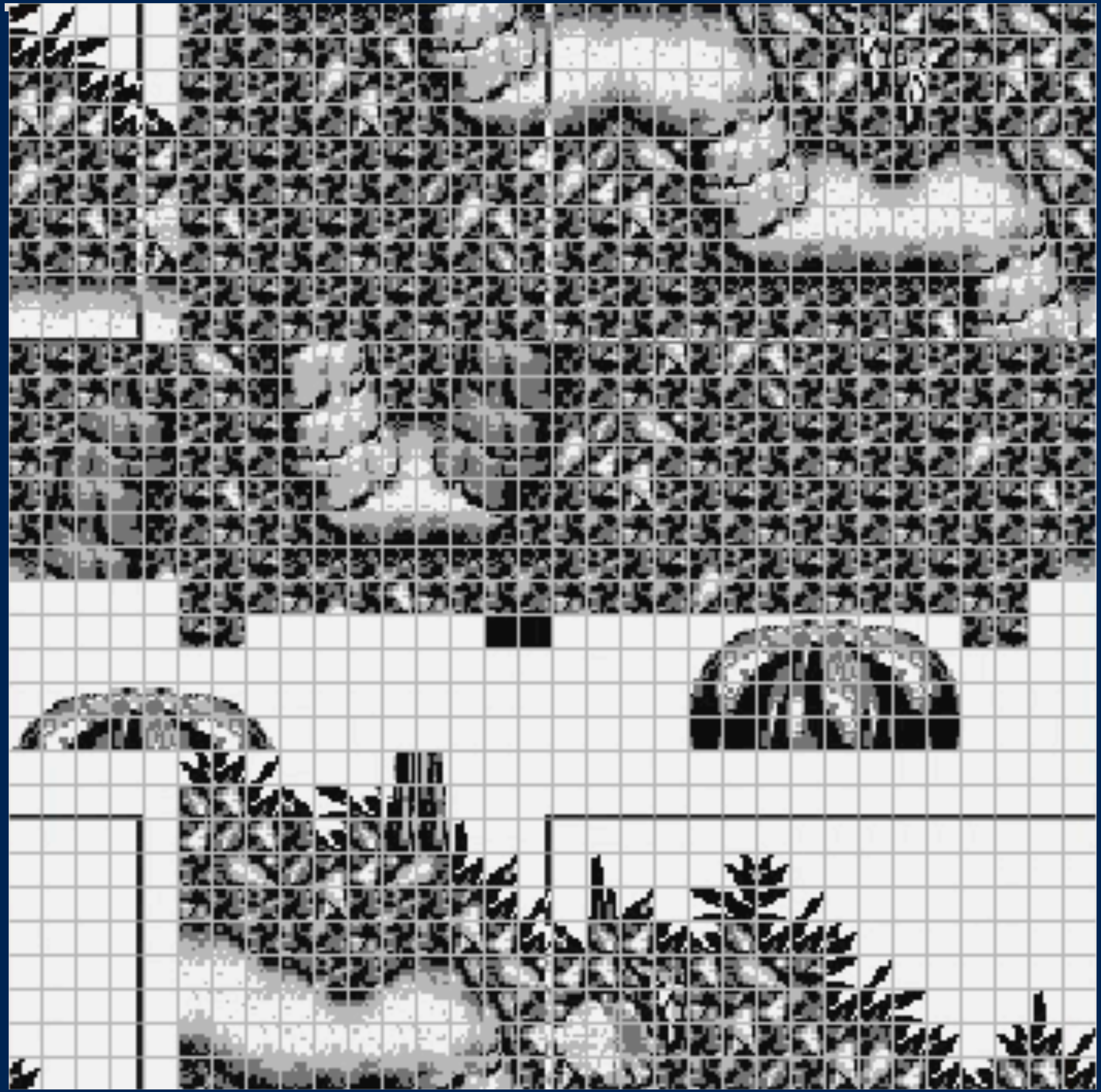

MARIO×02 WORLD TIME
0 ×00 1-2 400

MARIO×02 WORLD TIME
0 ×00 1-2 400


MARIO×02 WORLD TIME
0 ×00 1-2 400

MARIO×02 WORLD TIME
0 ×00 1-2 400







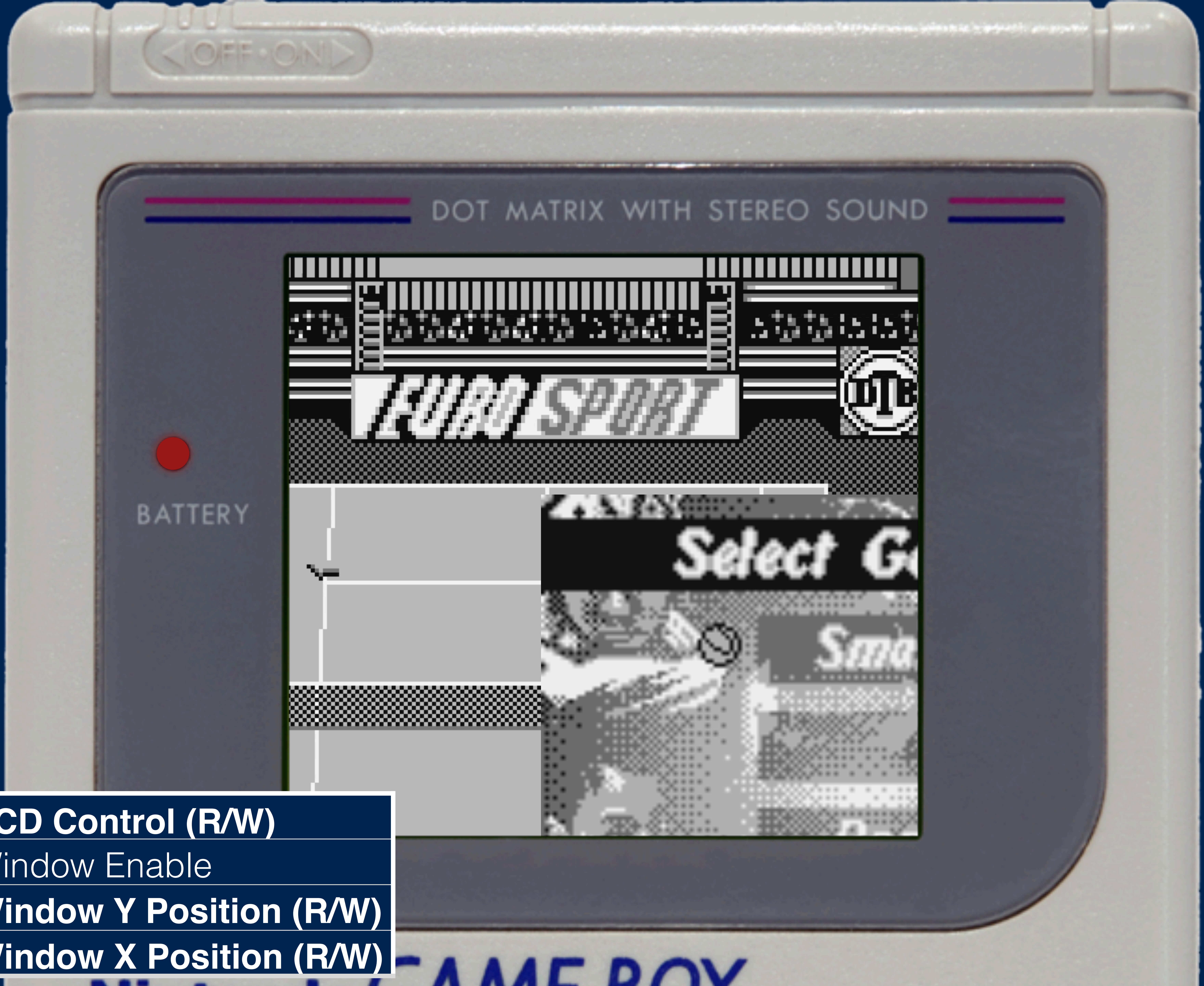



OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
EUROSPORT
FRANCE


OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
EURO SPORT
FRANCE


OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Select Game
Smash Tennis
Arcade
Bomb Tennis
Options
All Star Tennis

Window

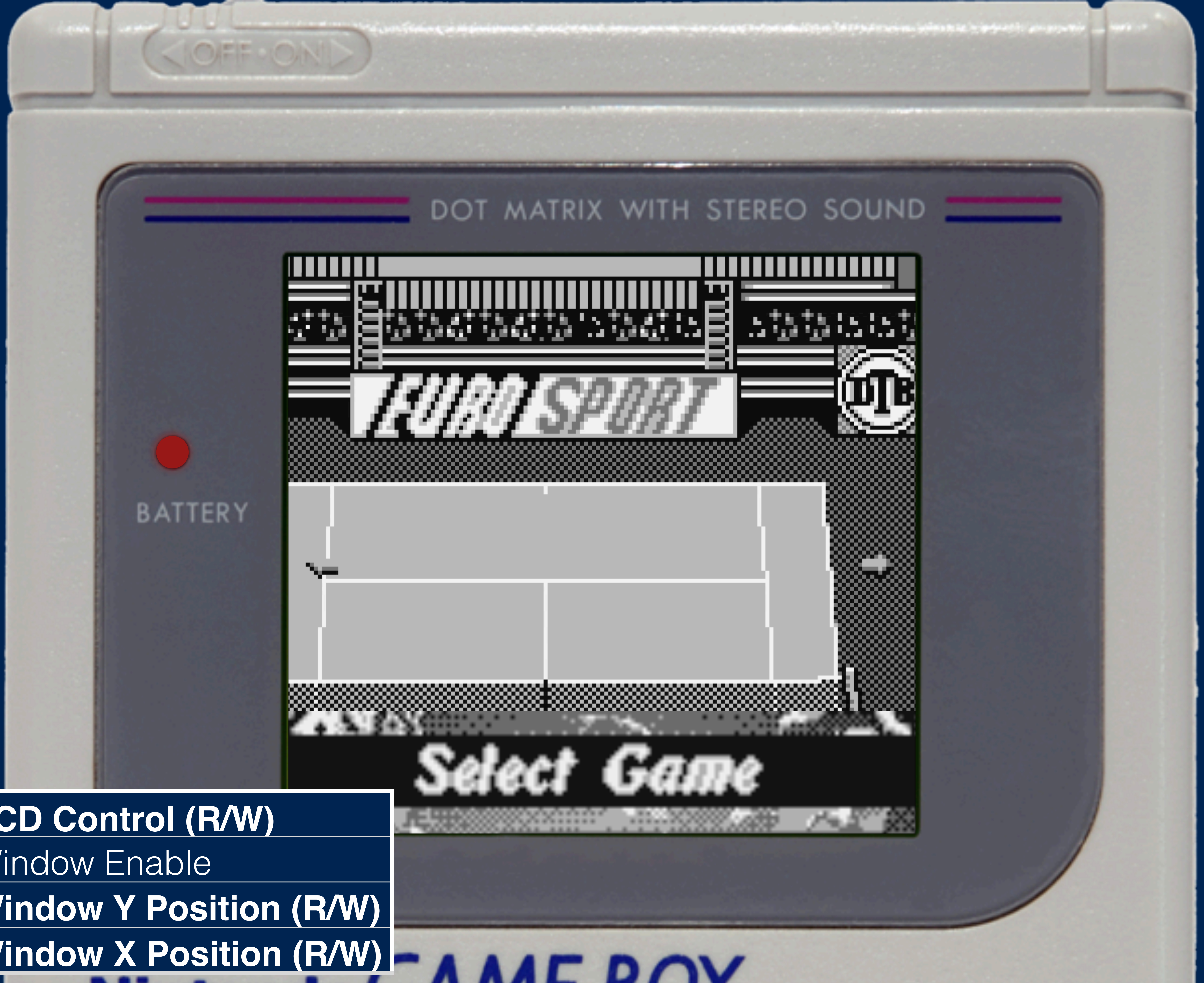


FF40	LCDC	LCD Control (R/W)
5		Window Enable
FF4A	WY	Window Y Position (R/W)
FF4B	WX	Window X Position (R/W)

Window

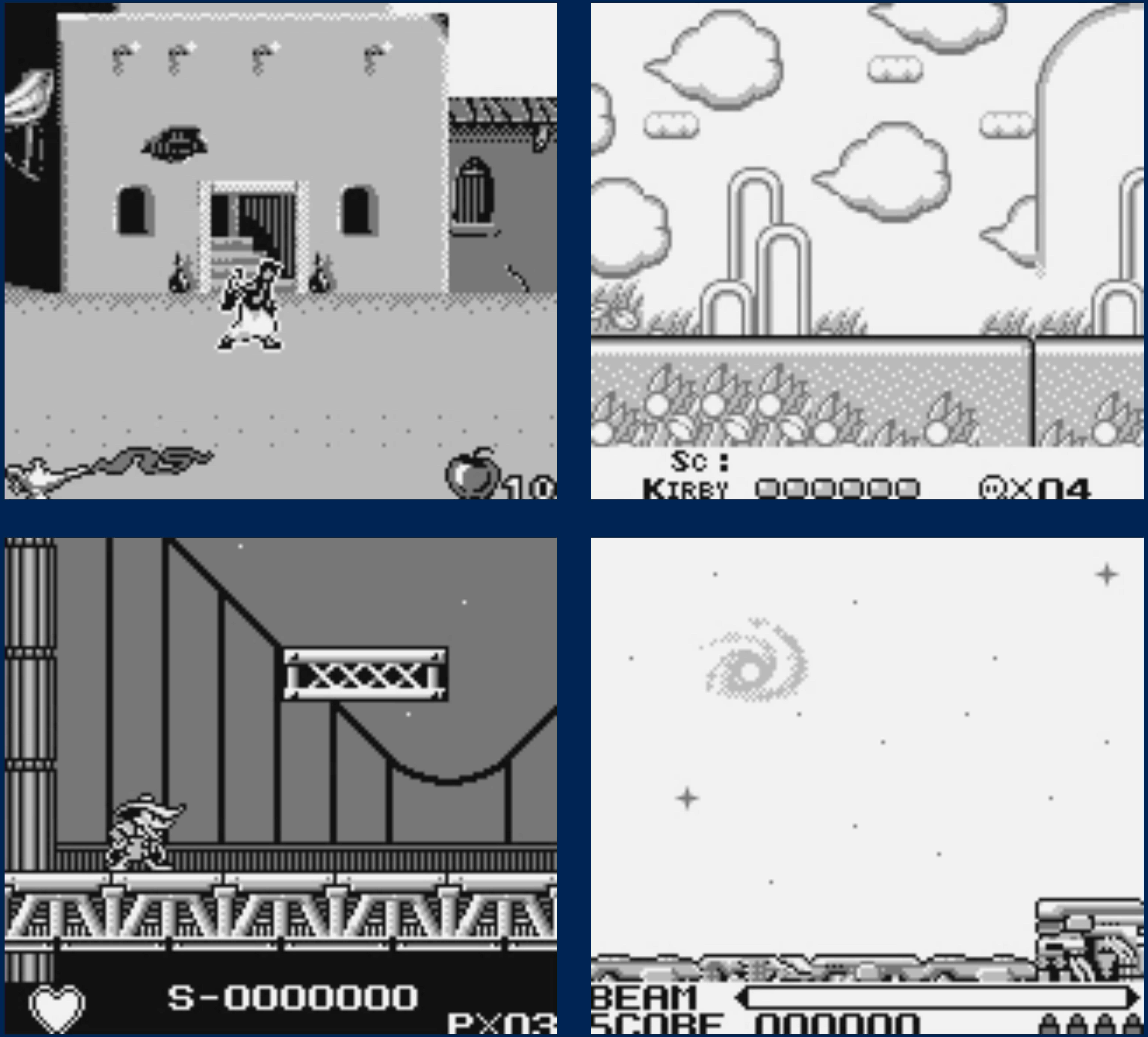
FF40	LCDC	**LCD Control (R/W)**
5		Window Enable
FF4A	WY	**Window Y Position (R/W)**
FF4B	WX	**Window X Position (R/W)**

Window

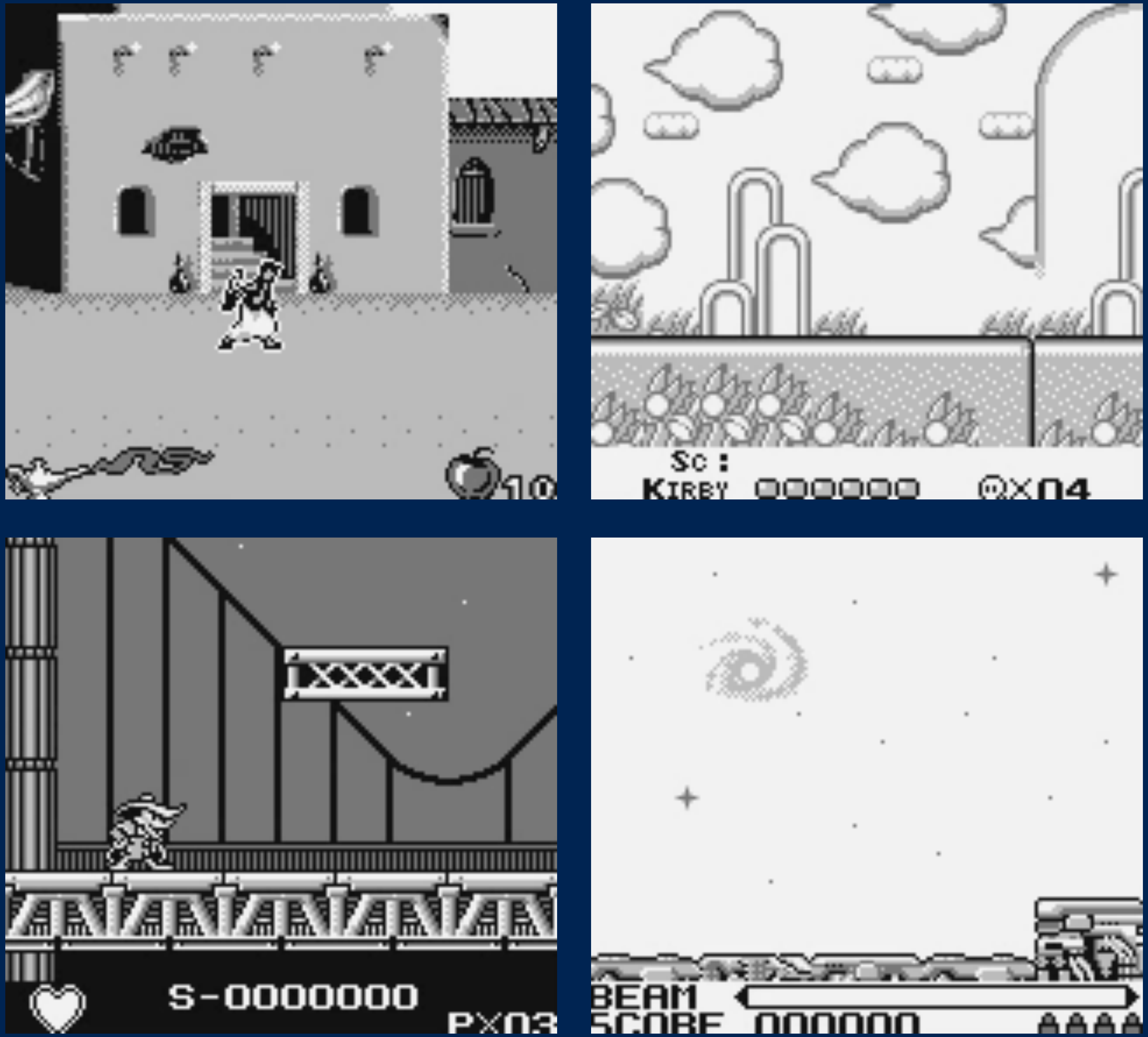


FF40	LCDC	**LCD Control (R/W)**
5		Window Enable
FF4A	WY	**Window Y Position (R/W)**
FF4B	WX	**Window X Position (R/W)**

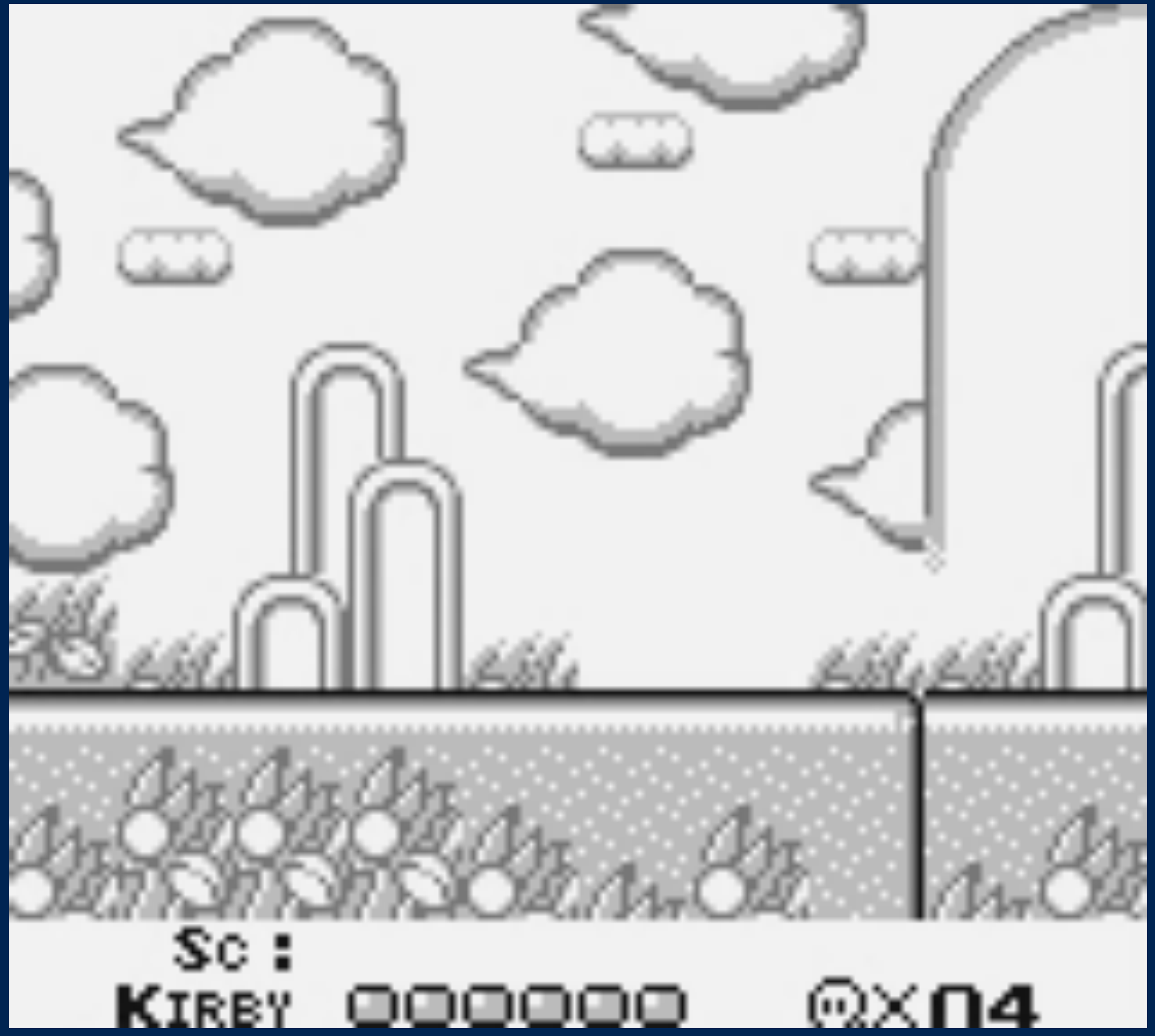
Window

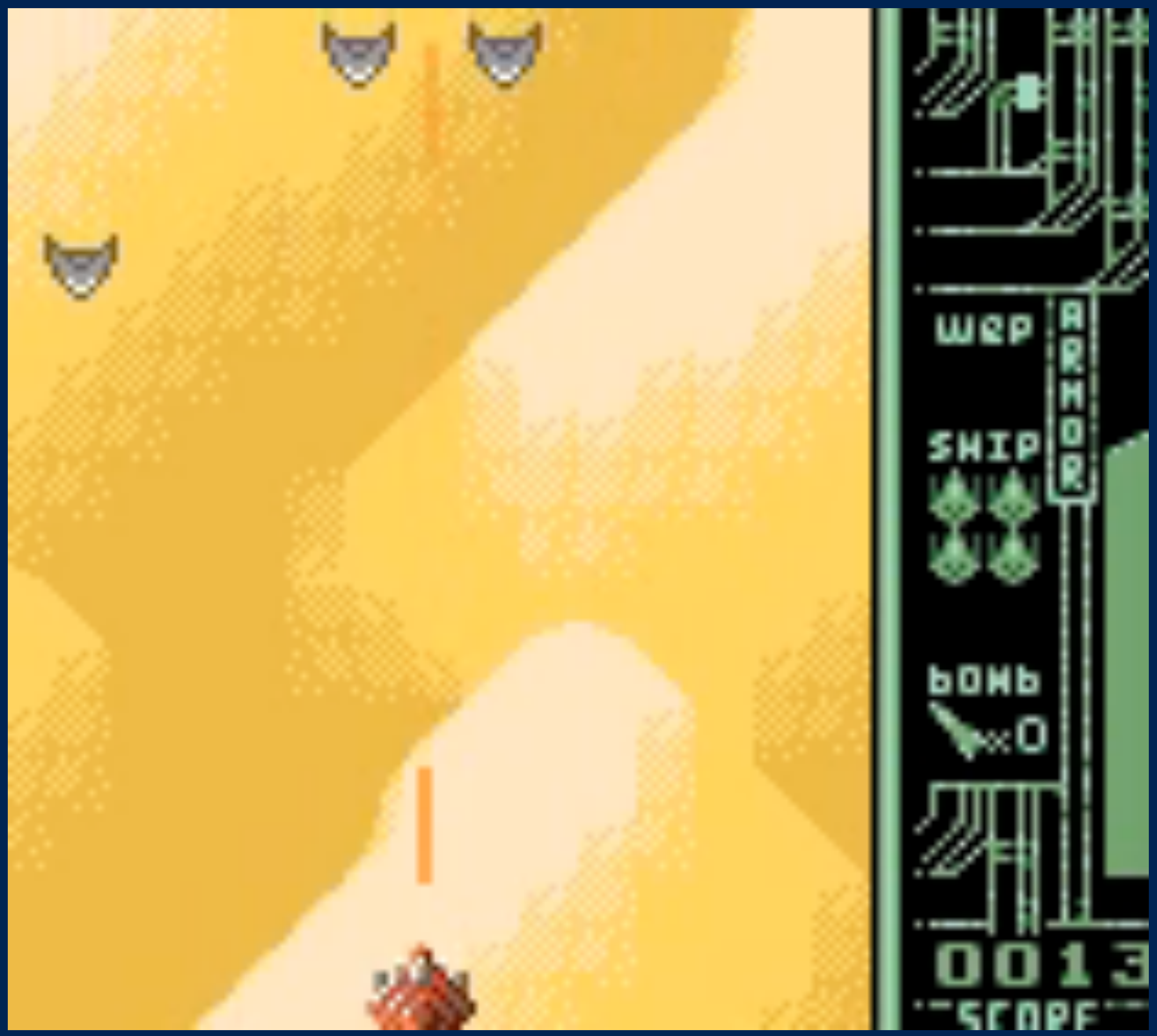


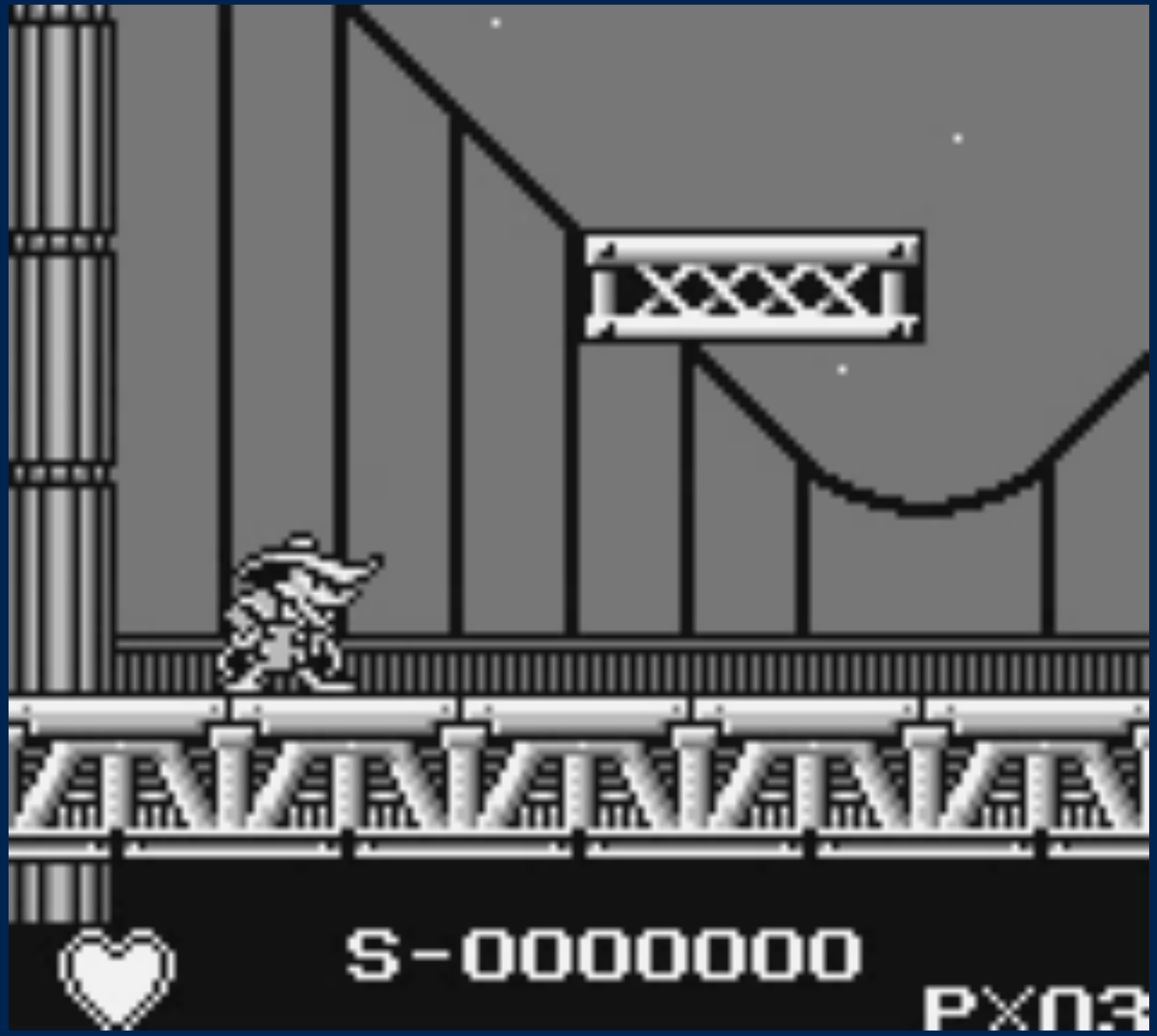
Window



Window

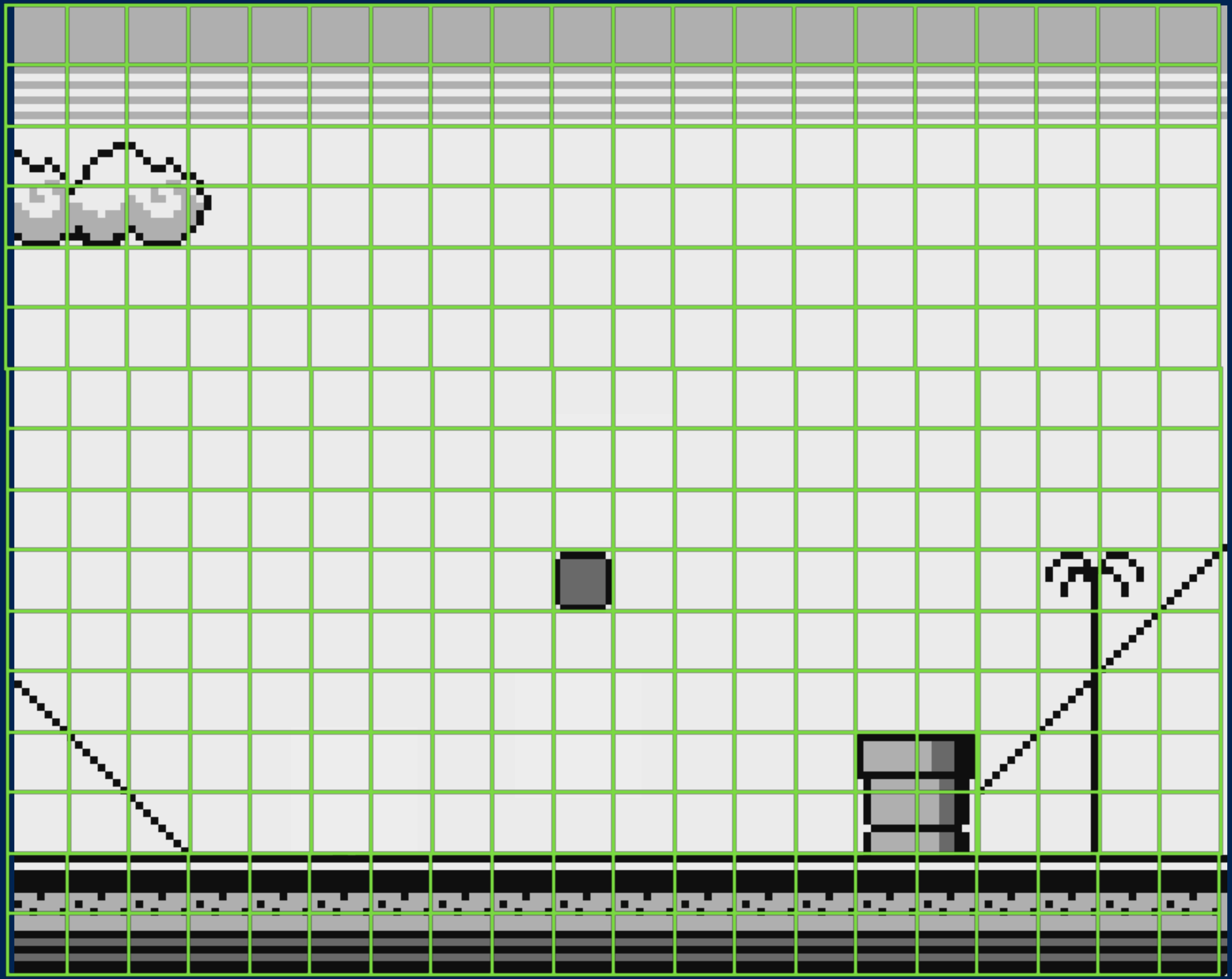

Sc :
KIRBY

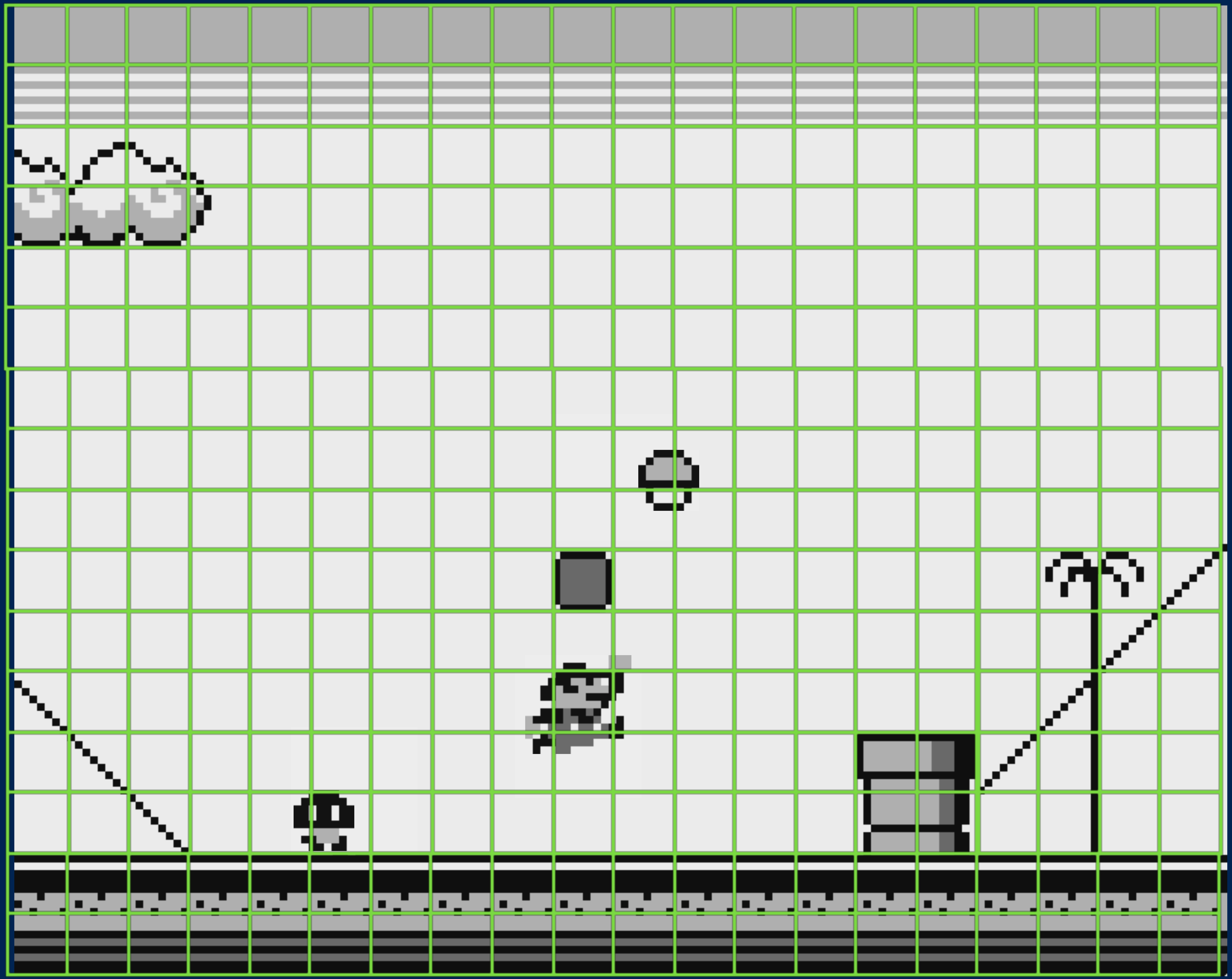

wep
ARMOR
SHIP
bOMb
0013
SCORE


S-0000000

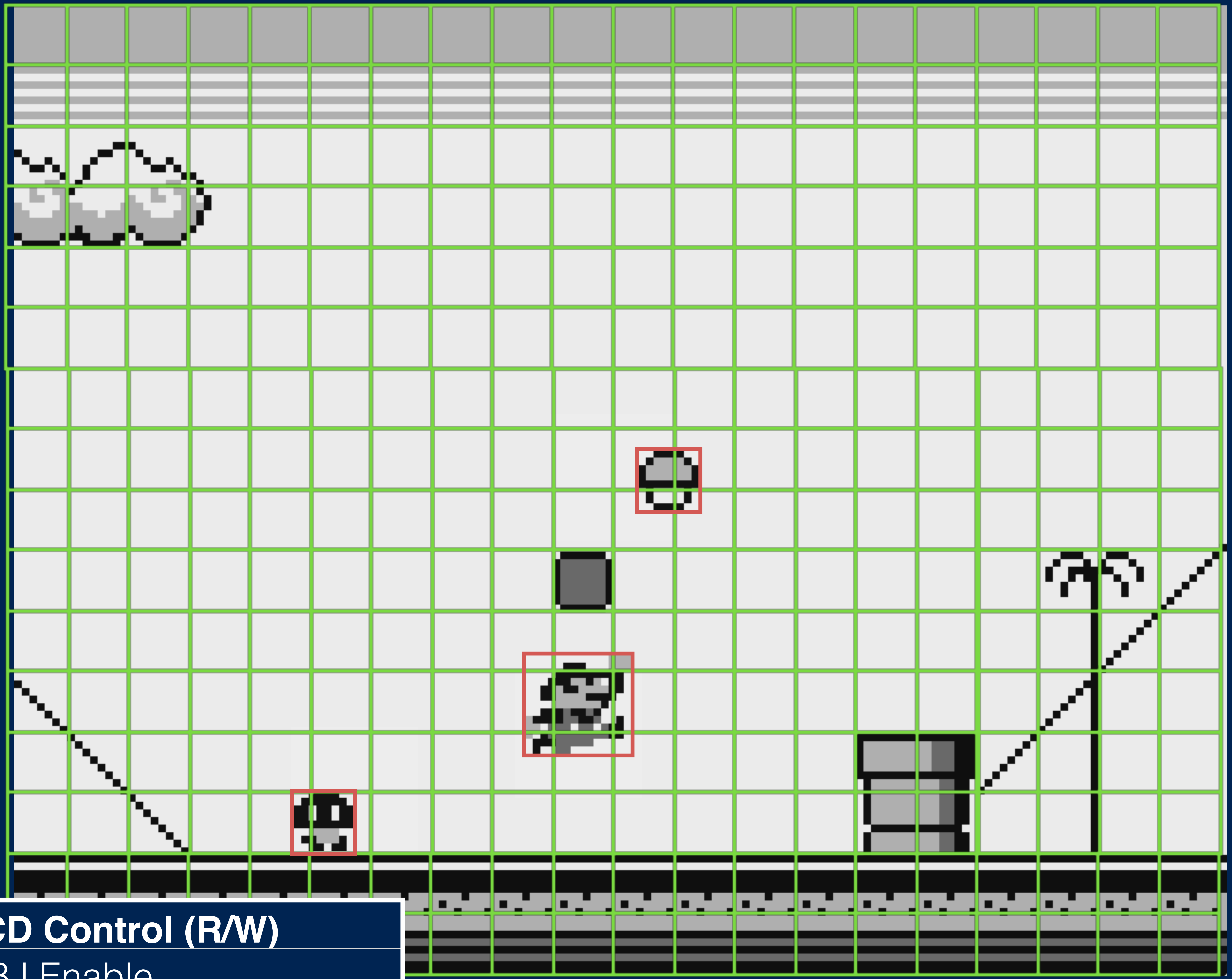

BEAM
SCORE 000000


NO
ITEM
NO
ITEM
LIFE



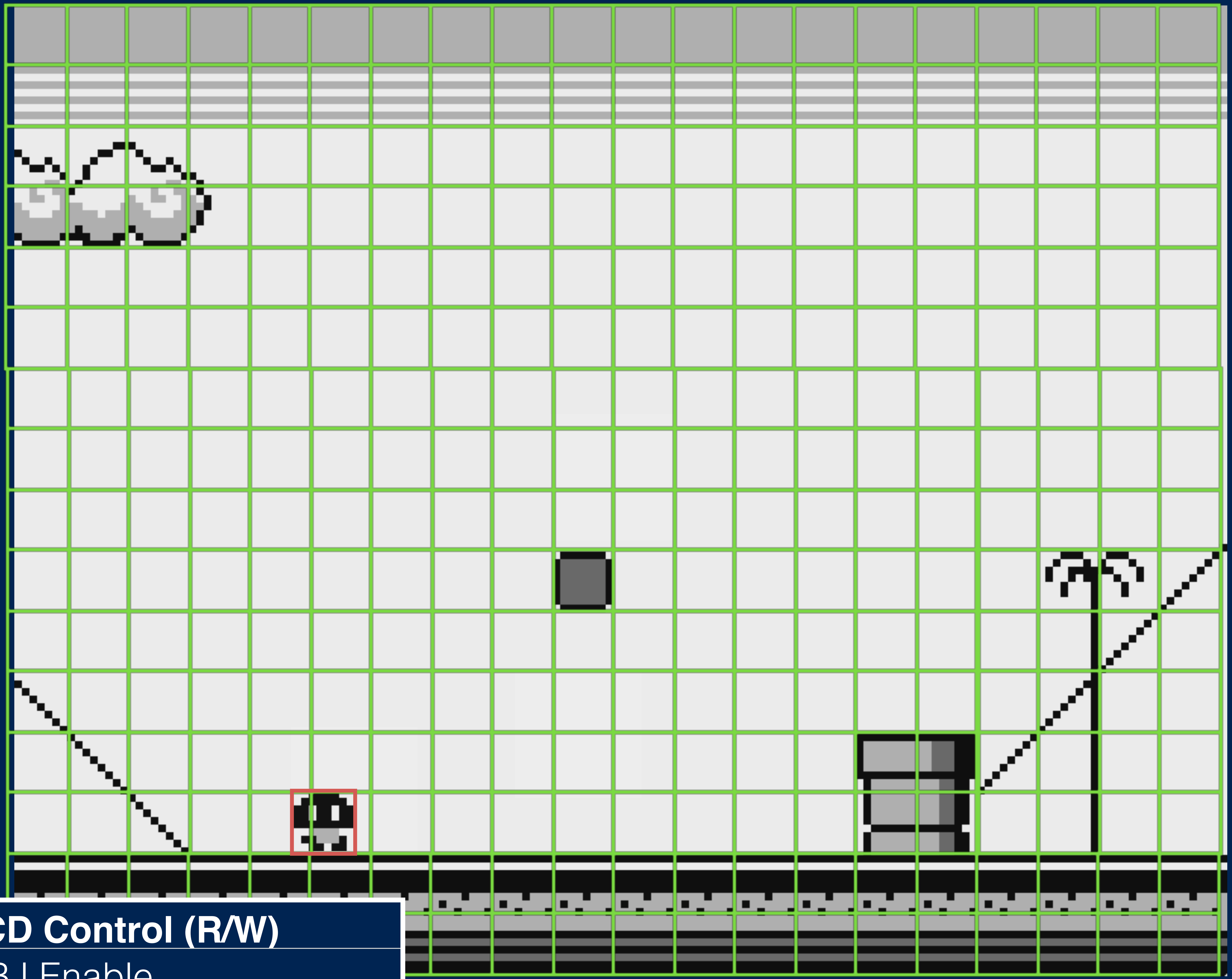


Sprites



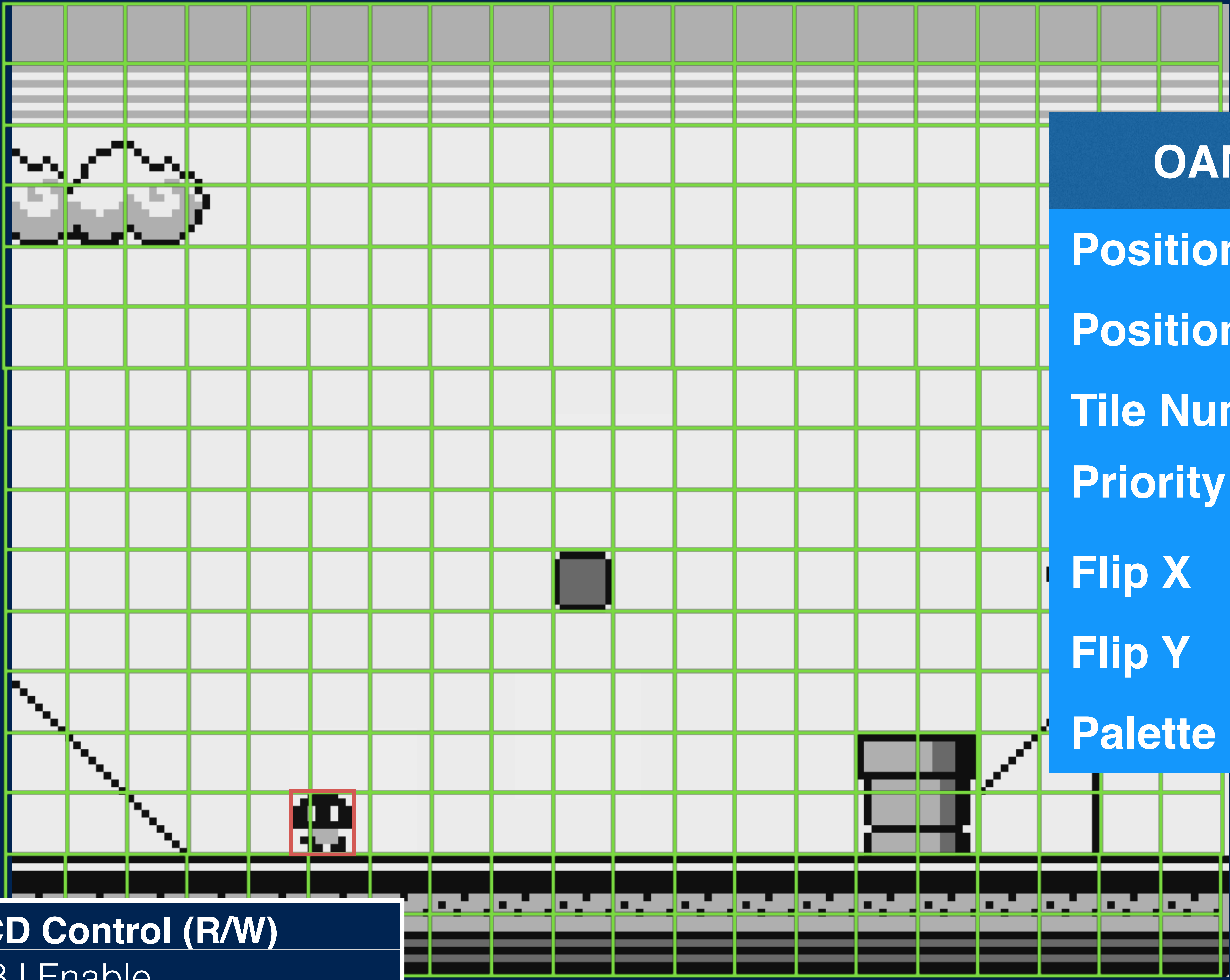
FF40	LCDC	LCD Control (R/W)
1		OBJ Enable

Sprites



FF40	LCDC	LCD Control (R/W)
1		OBJ Enable

Sprites

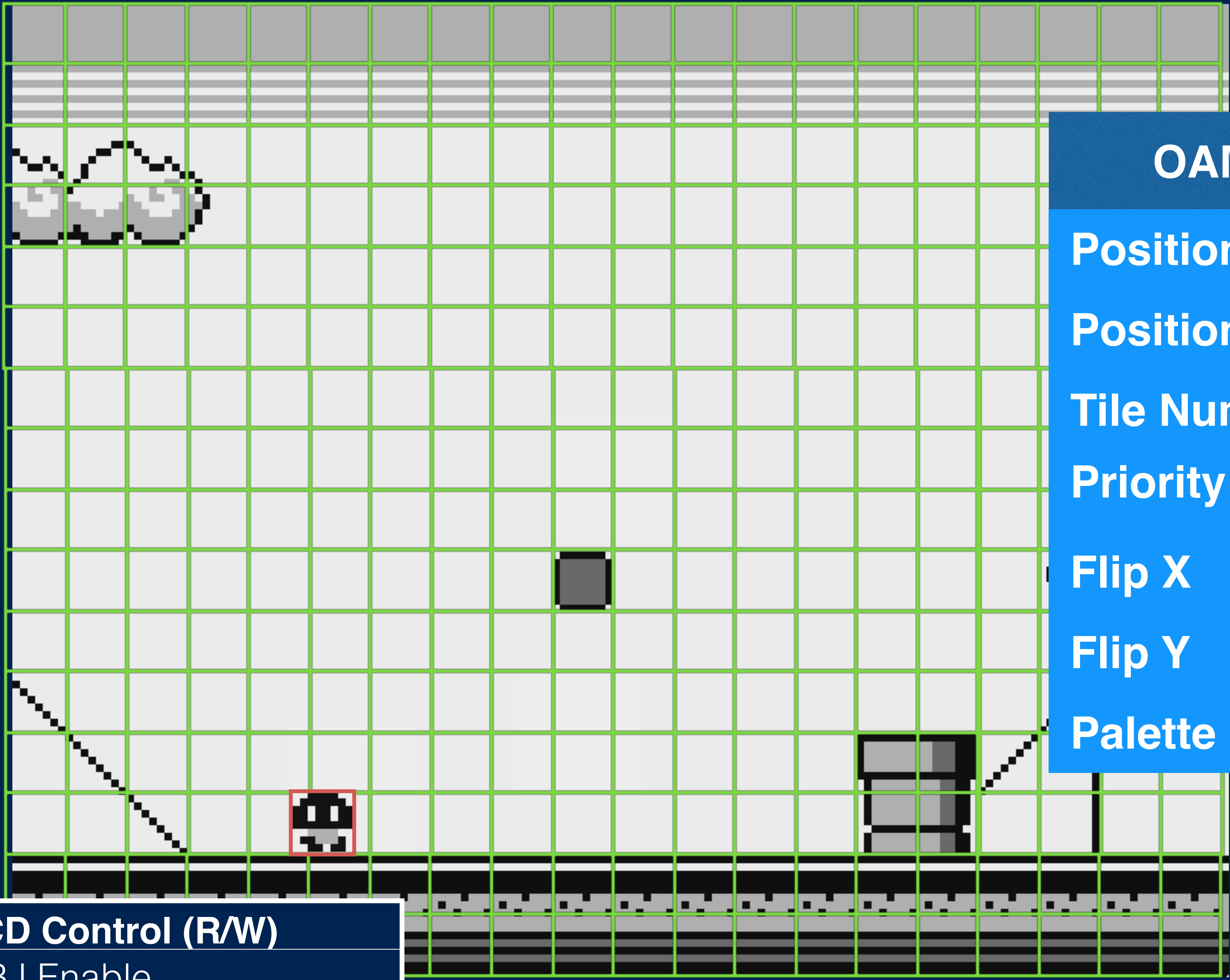


OAM Entry

Position X	0x__
Position Y	0x__
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

FF40	LCDC	LCD Control (R/W)
1		OBJ Enable

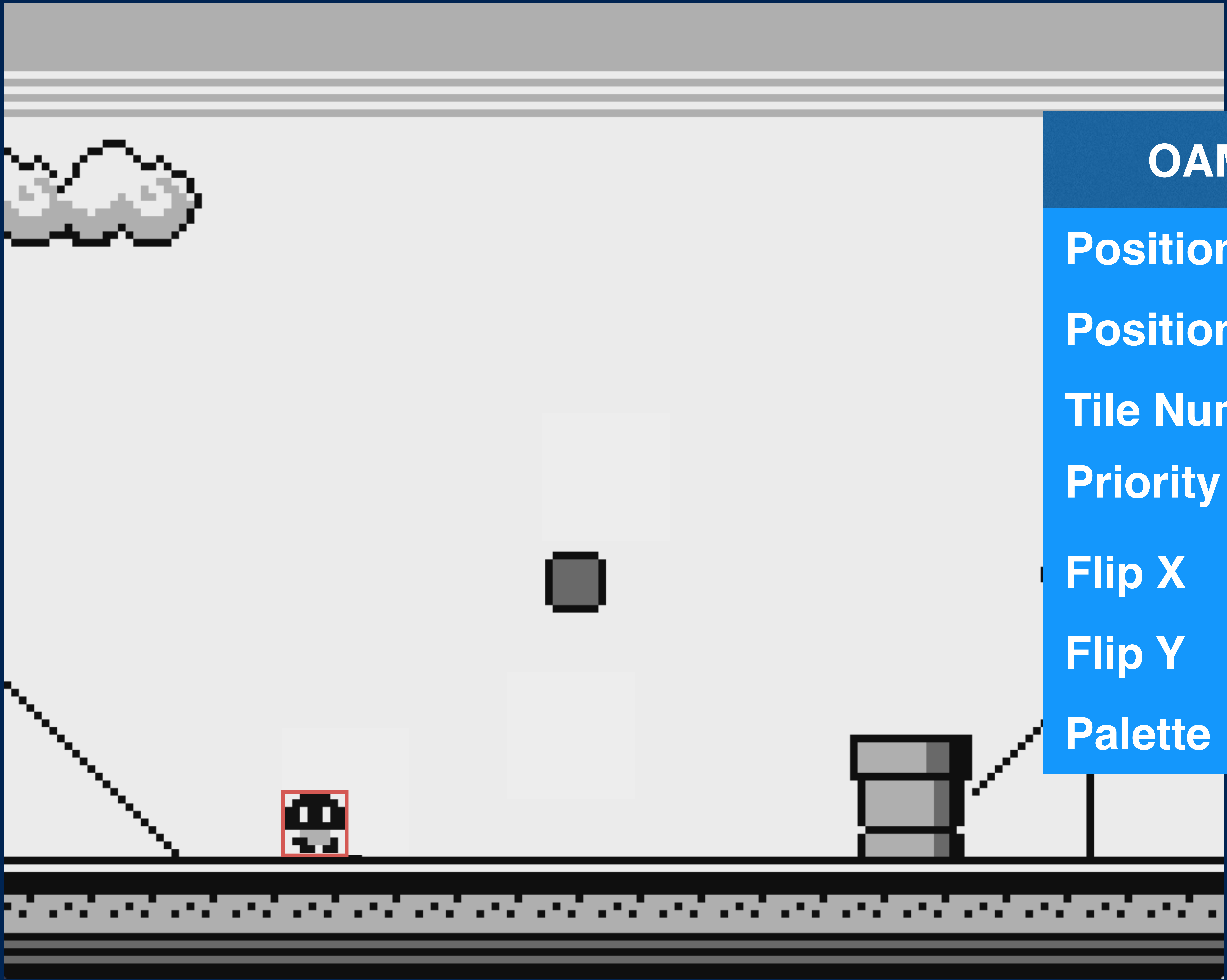
Sprites



OAM Entry

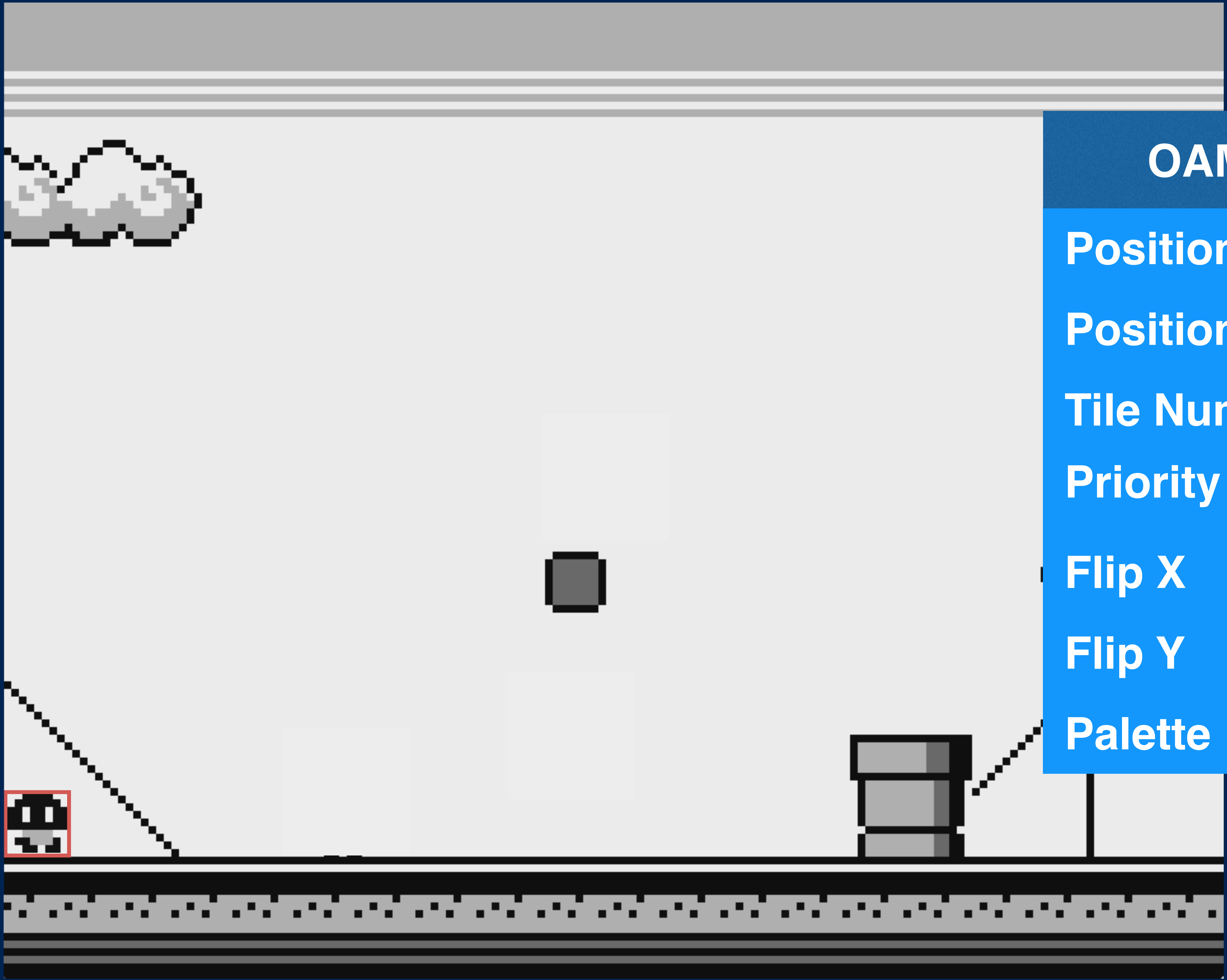
Position X	0x__
Position Y	0x__
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

FF40	LCDC	LCD Control (R/W)
1		OBJ Enable

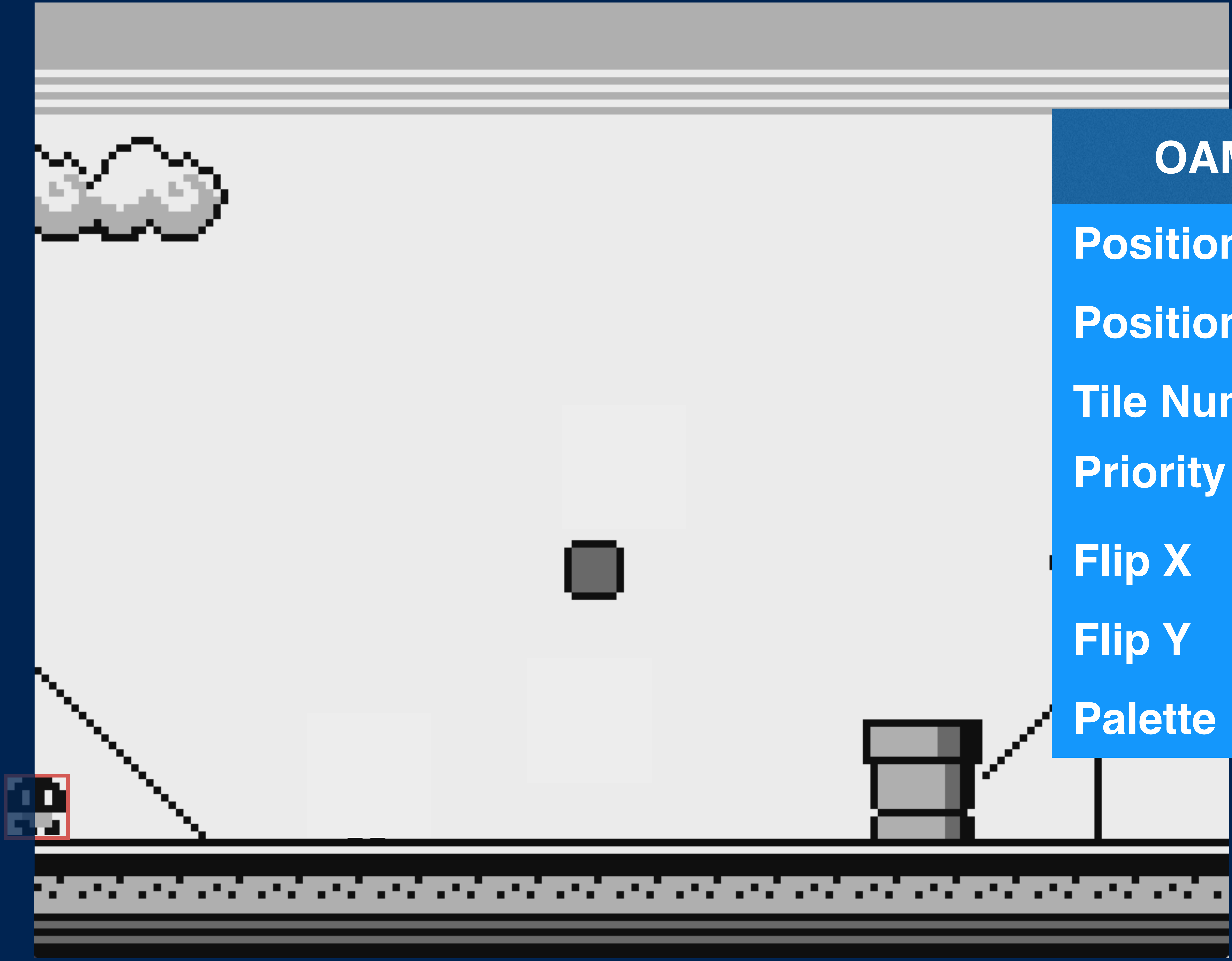


OAM Entry

Position X	0x__
Position Y	0x__
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

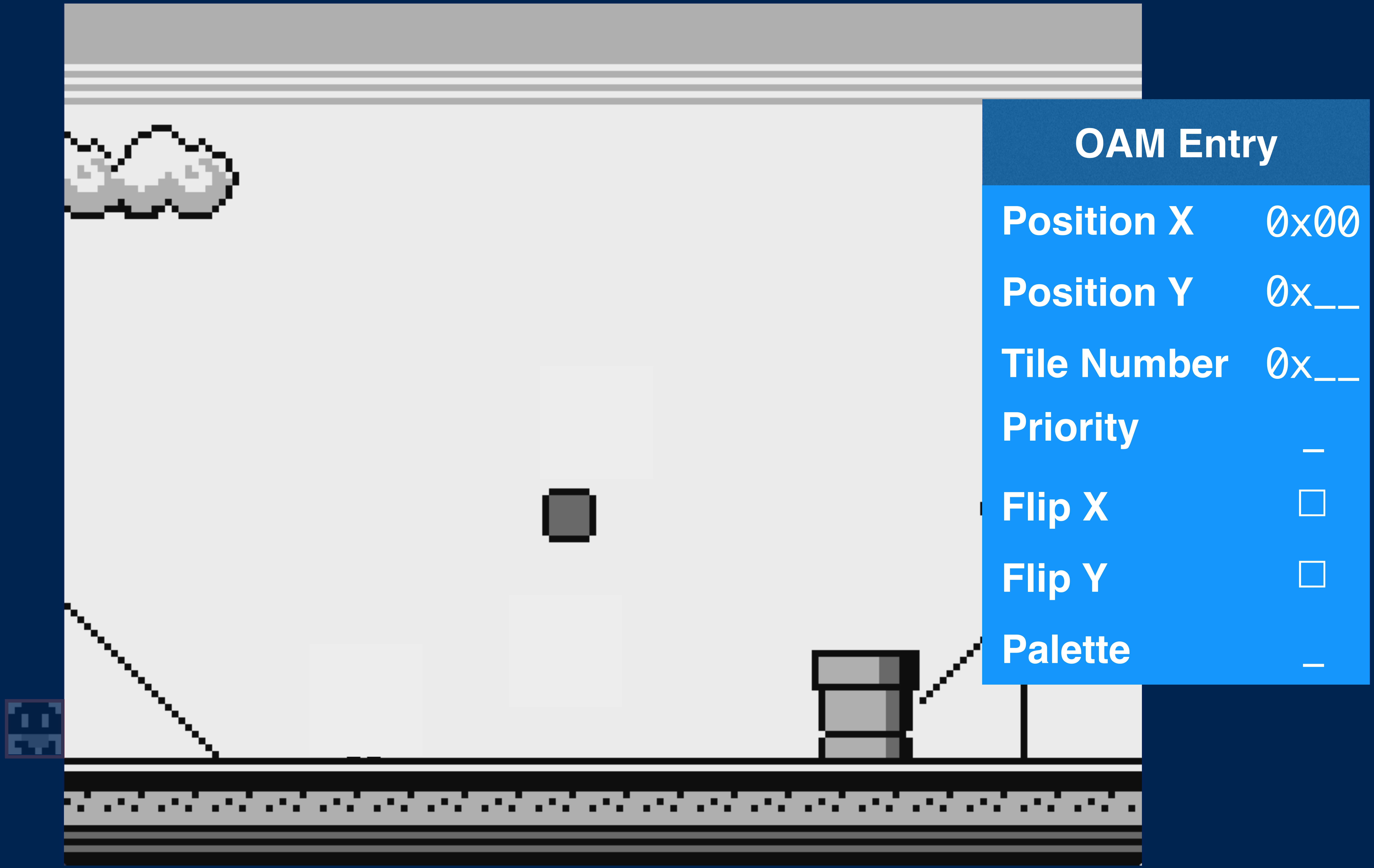


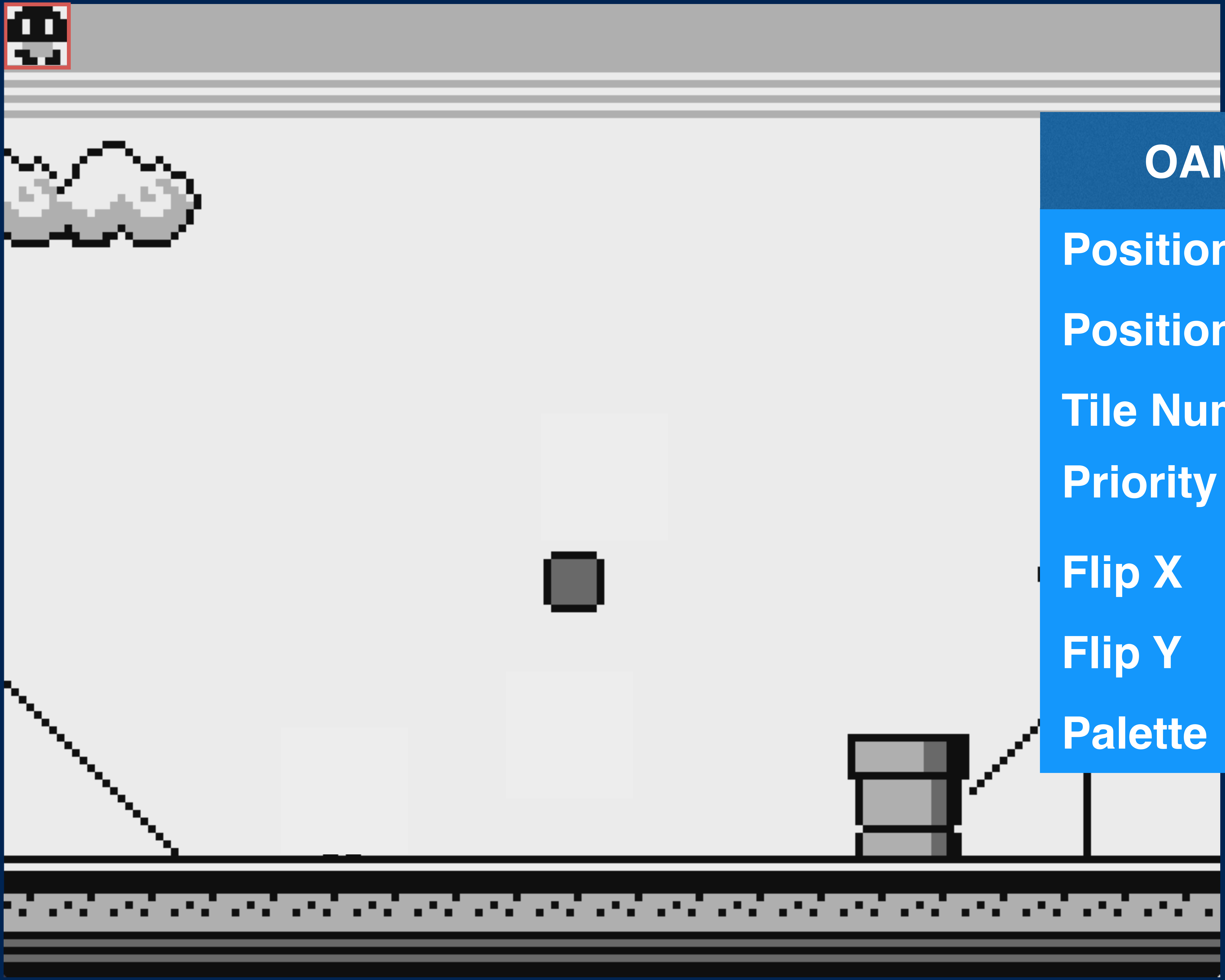
OAM Entry	
Position X	0x08
Position Y	0x__
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_



OAM Entry

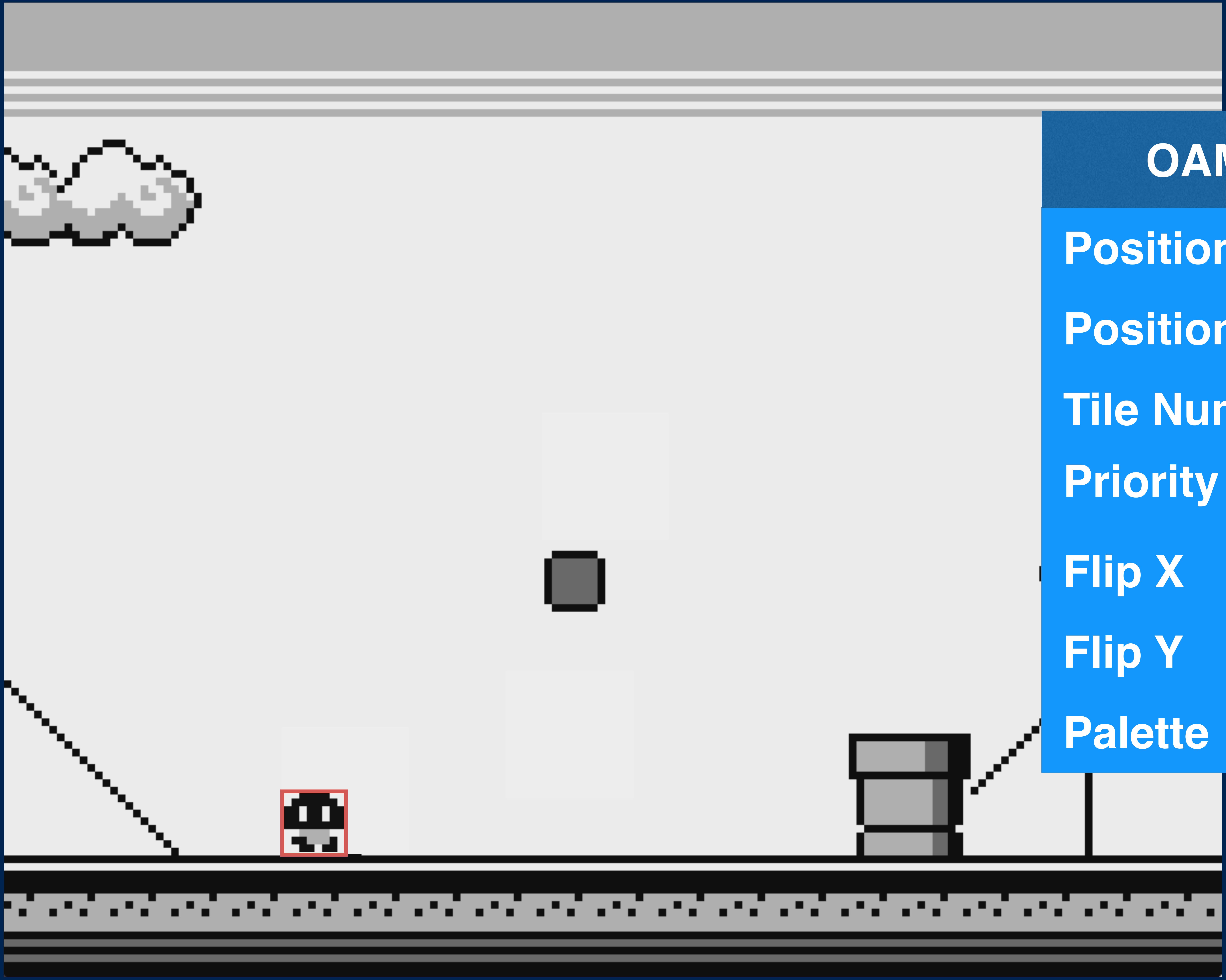
Position X	0x04
Position Y	0x__
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_


OAM Entry
Position X 0x00
Position Y 0x__
Tile Number 0x__
Priority _
Flip X
Flip Y
Palette _



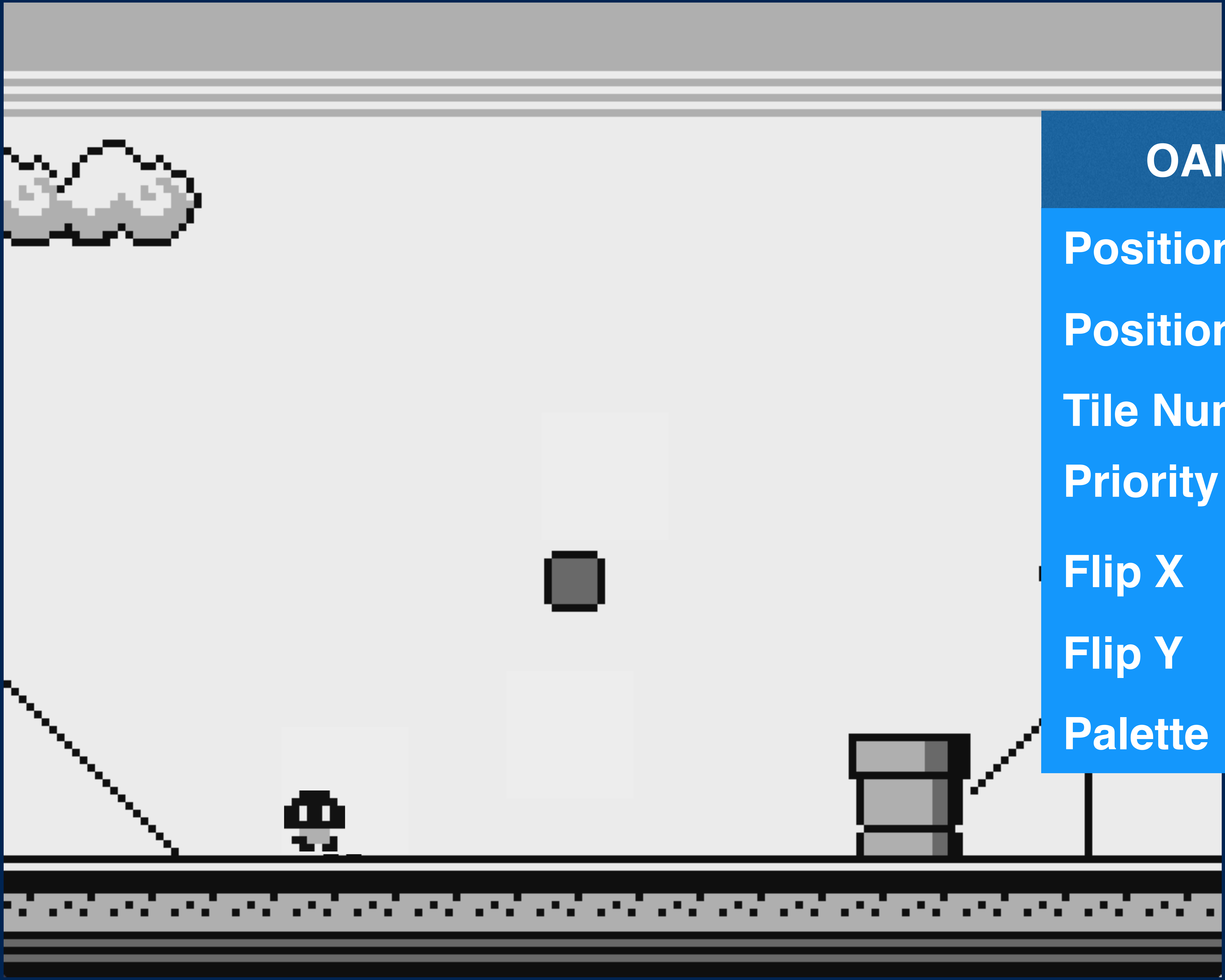
OAM Entry

Position X	0x08
Position Y	0x10
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_



OAM Entry

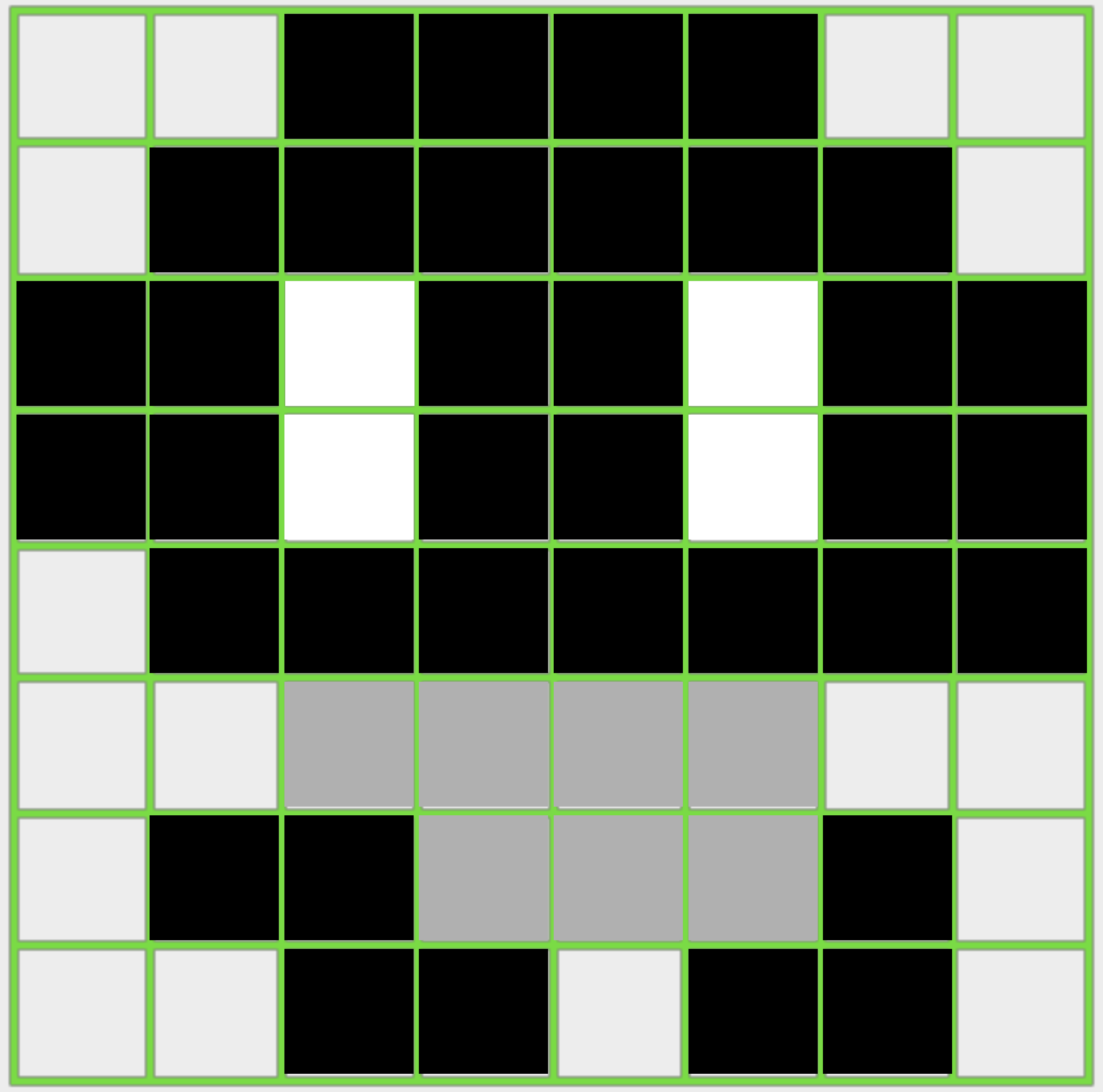
Position X	0x4D
Position Y	0x78
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_



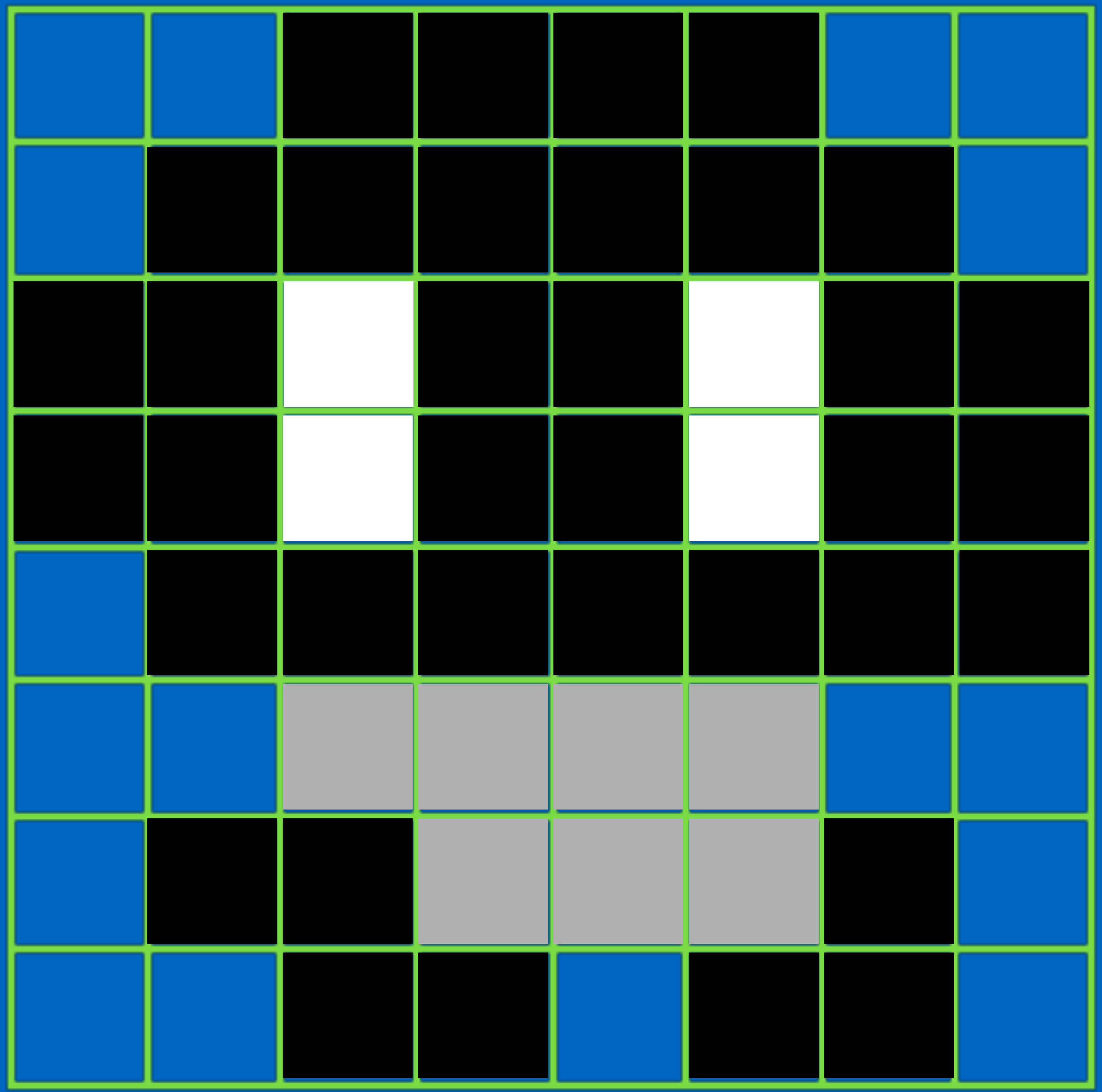
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

OAM Entry

Position X	0x4D
Position Y	0x78
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

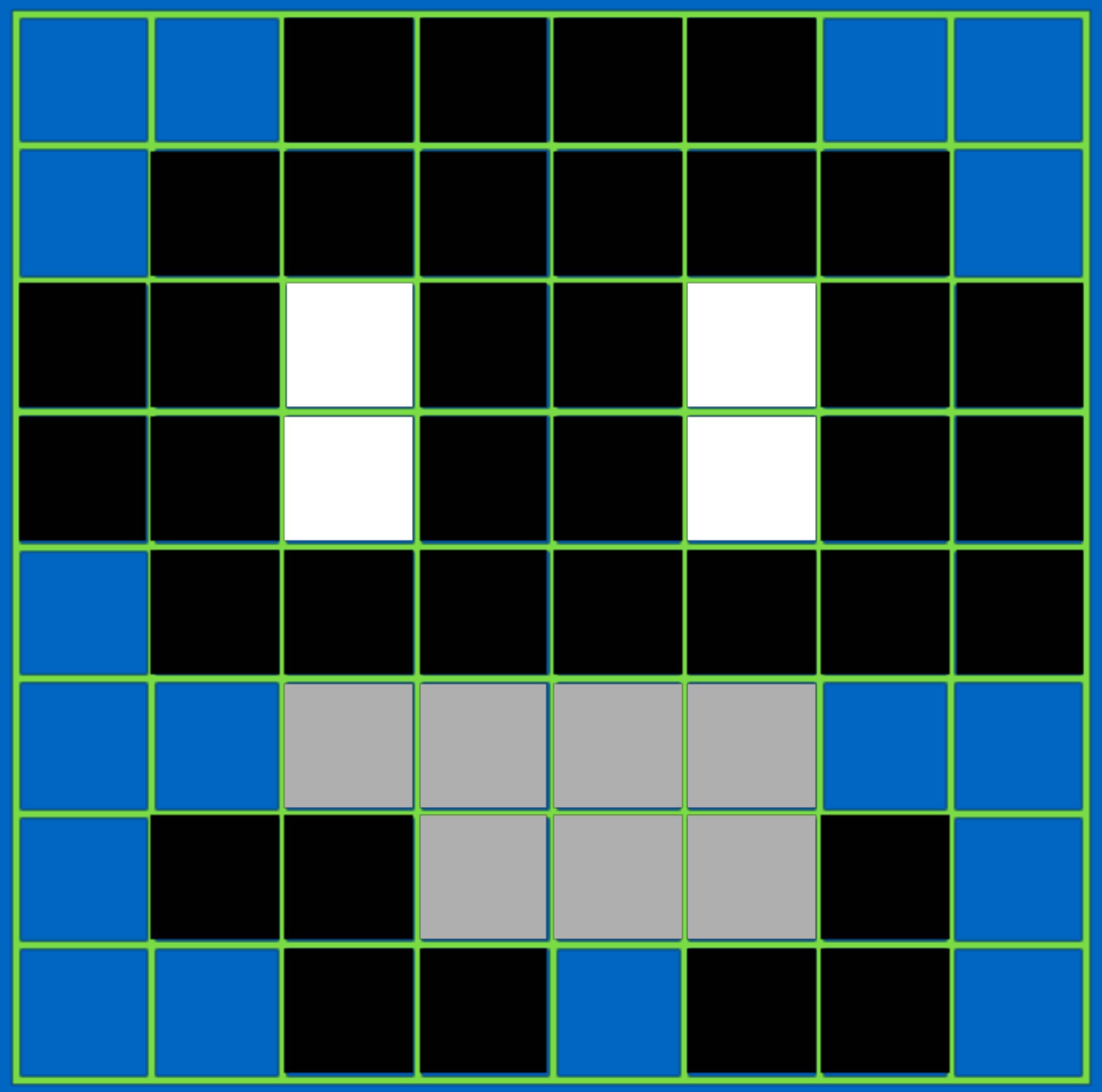


OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_



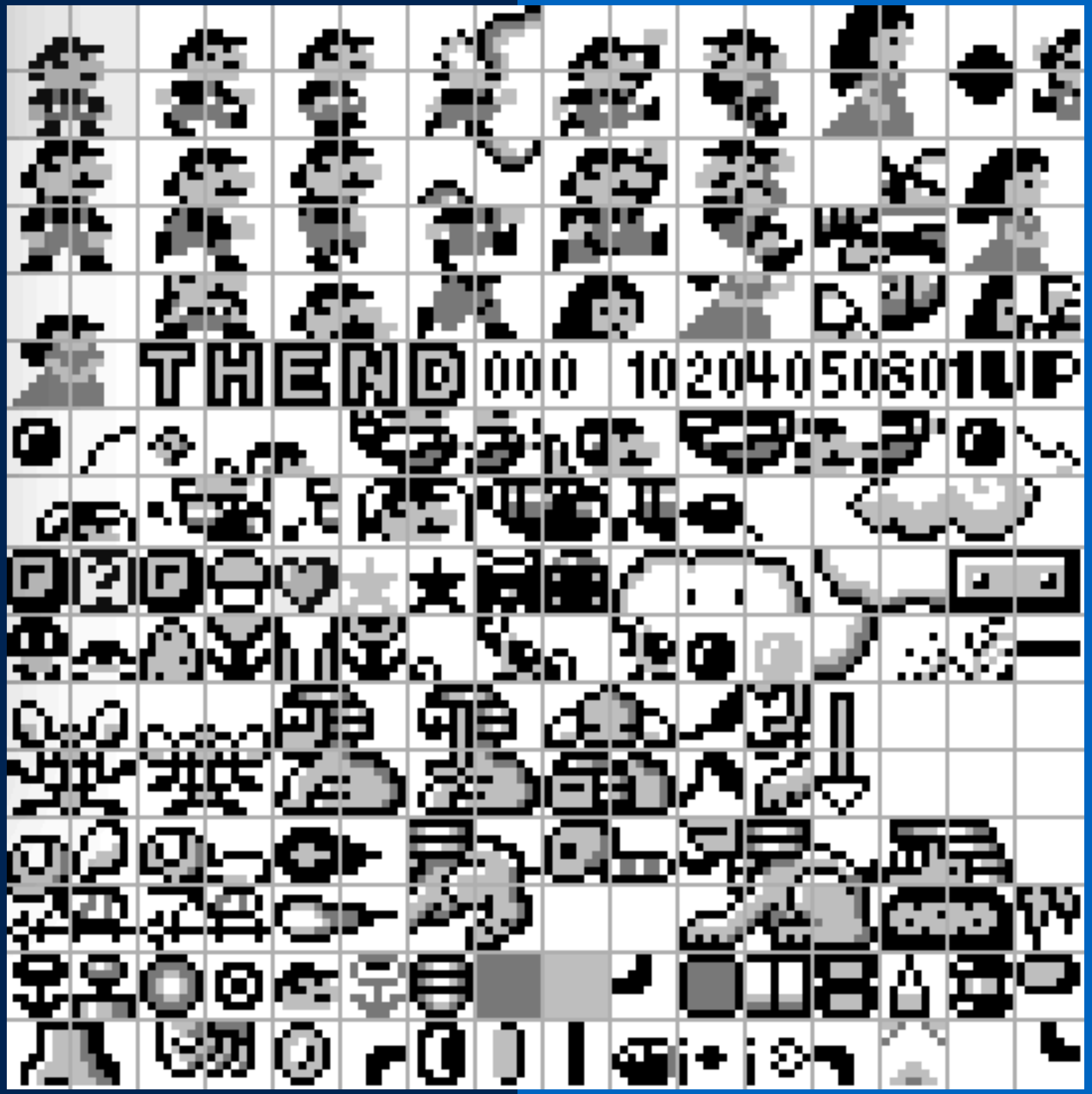
OAM Entry

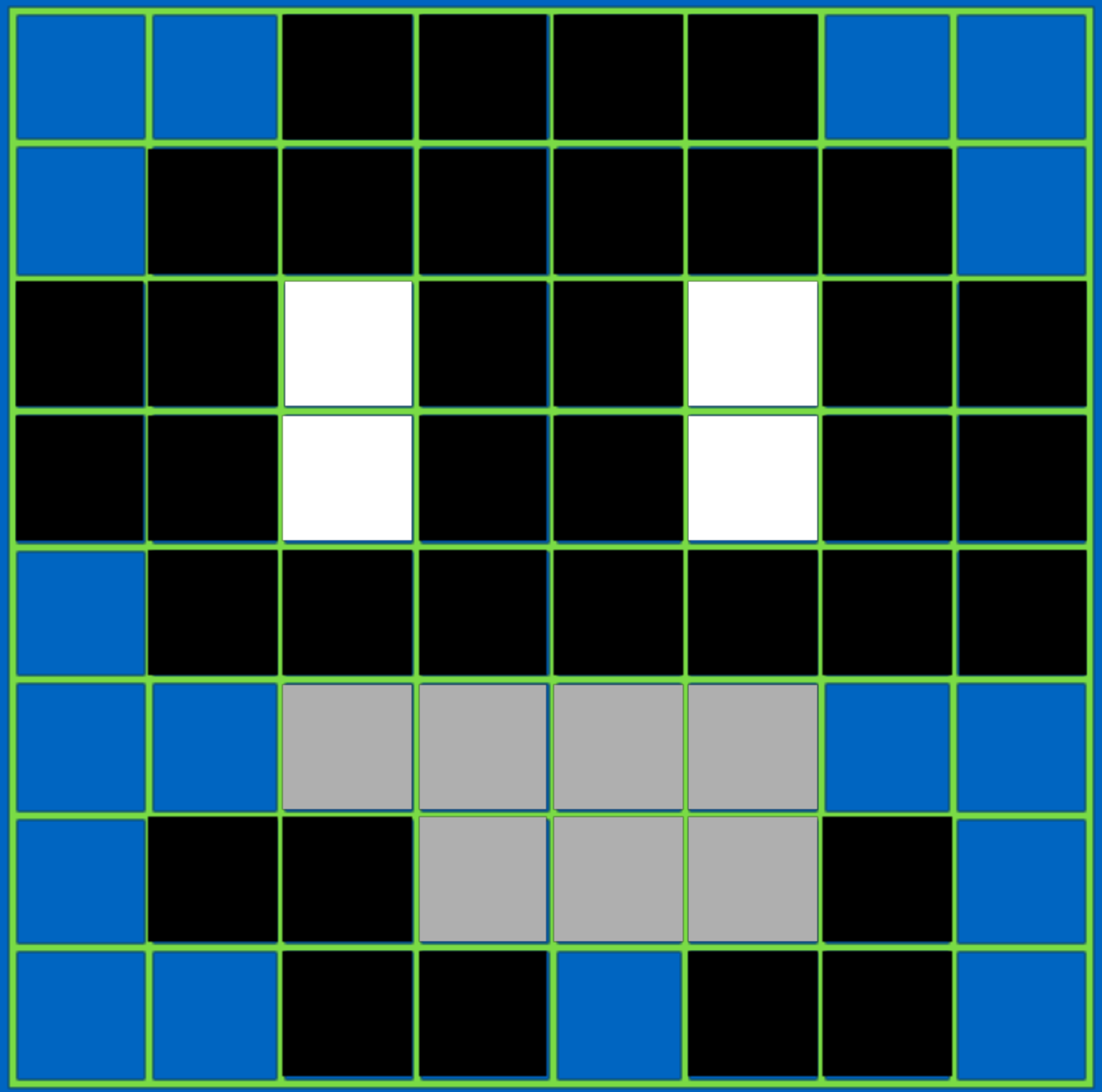
Position X	0x4D
Position Y	0x78
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_



OAM Entry

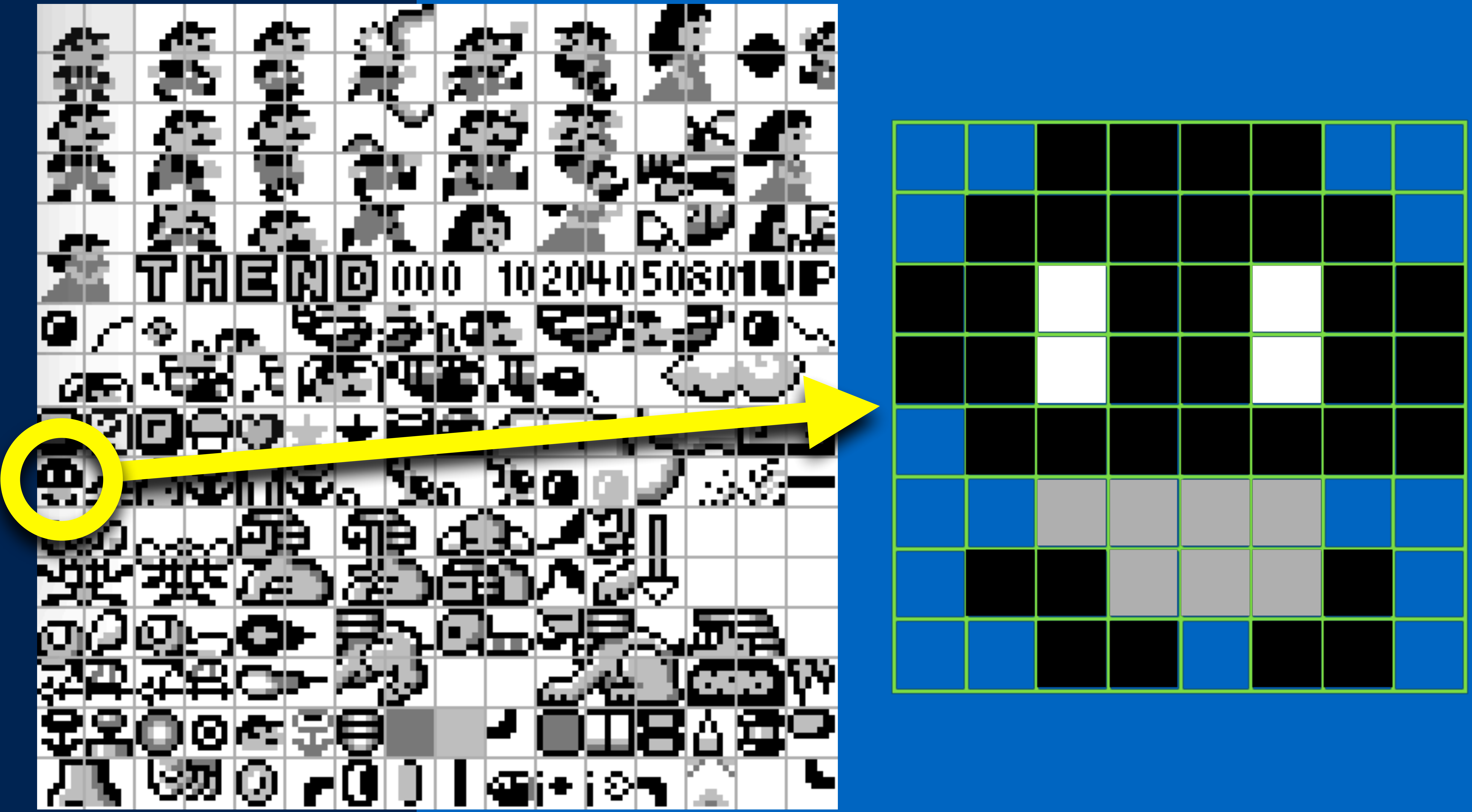
Position X	0x4D
Position Y	0x78
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_





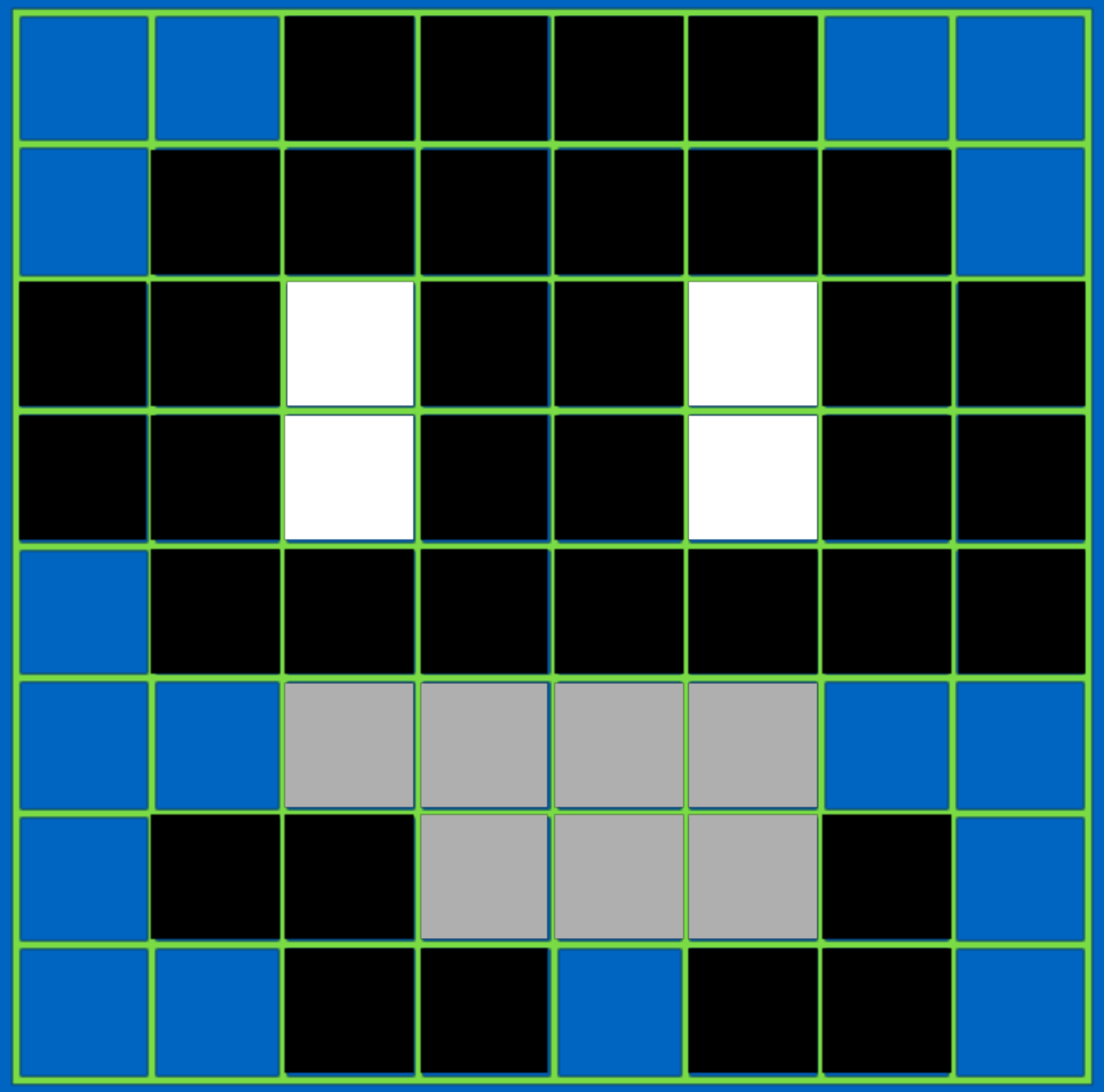
OAM Entry

Position X	0x4D
Position Y	0x78
Tile Number	0x__
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

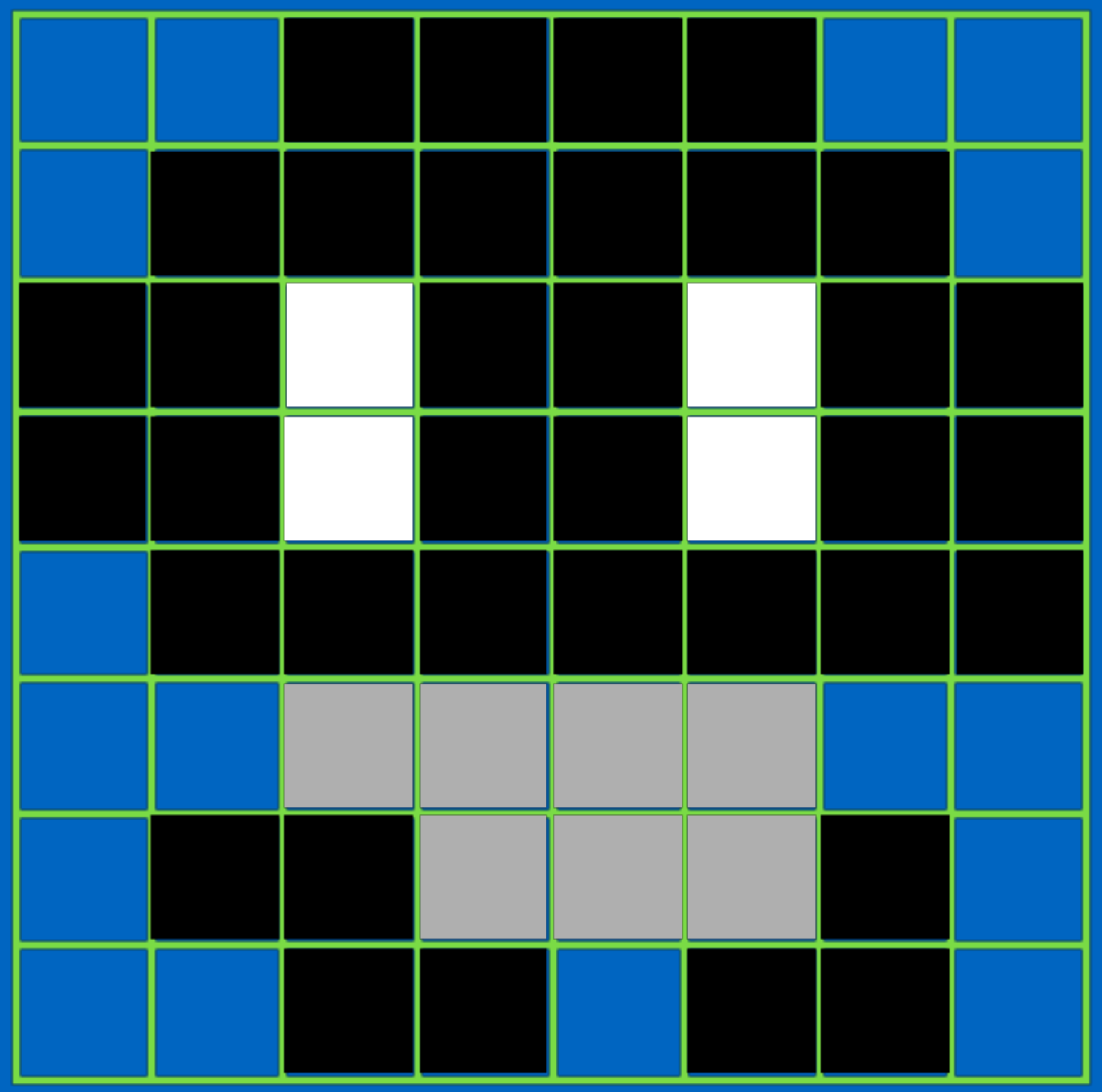


OAM Entry

Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

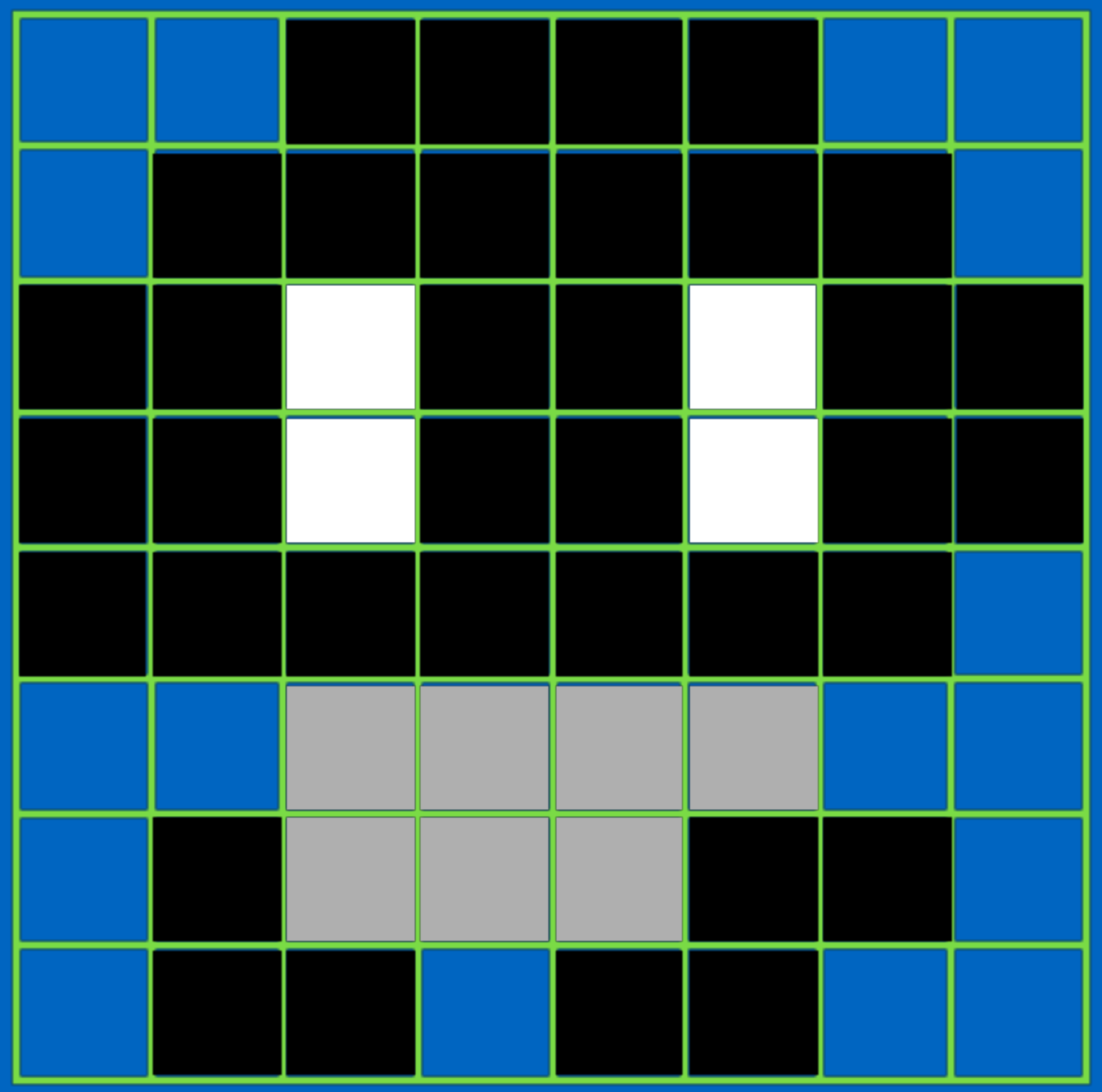


OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

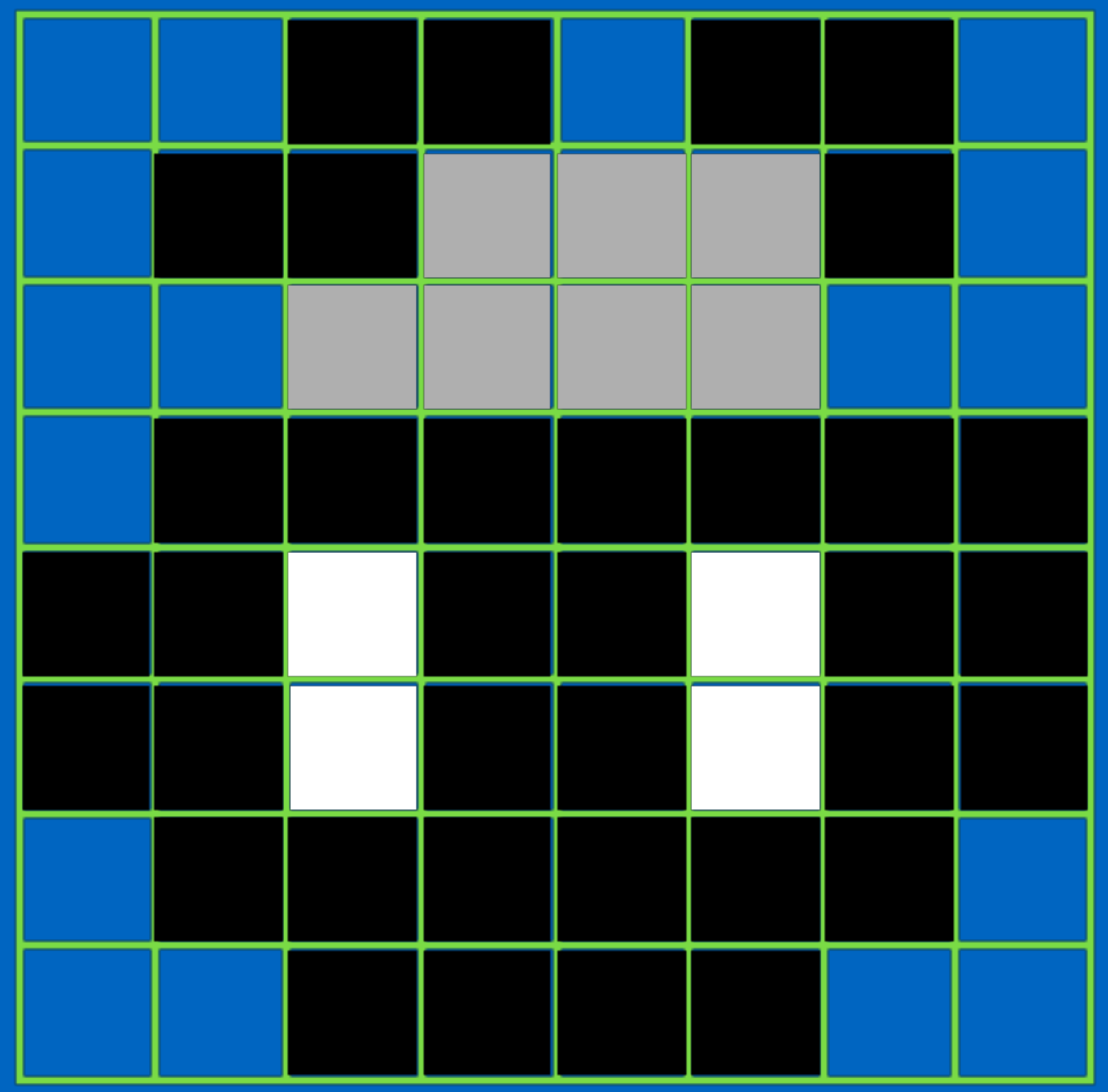


OAM Entry

Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

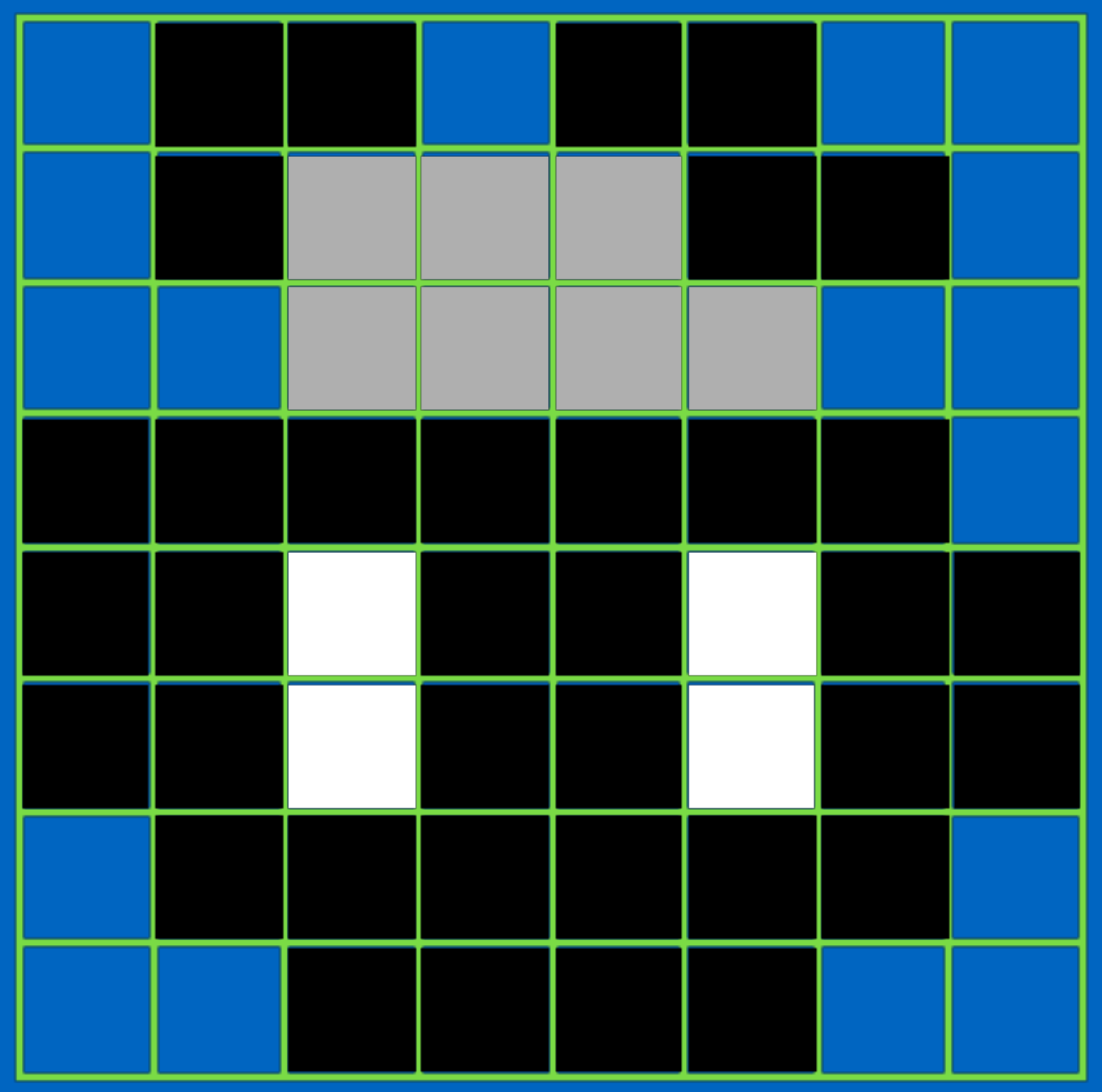


OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☒
Flip Y	☐
Palette	_

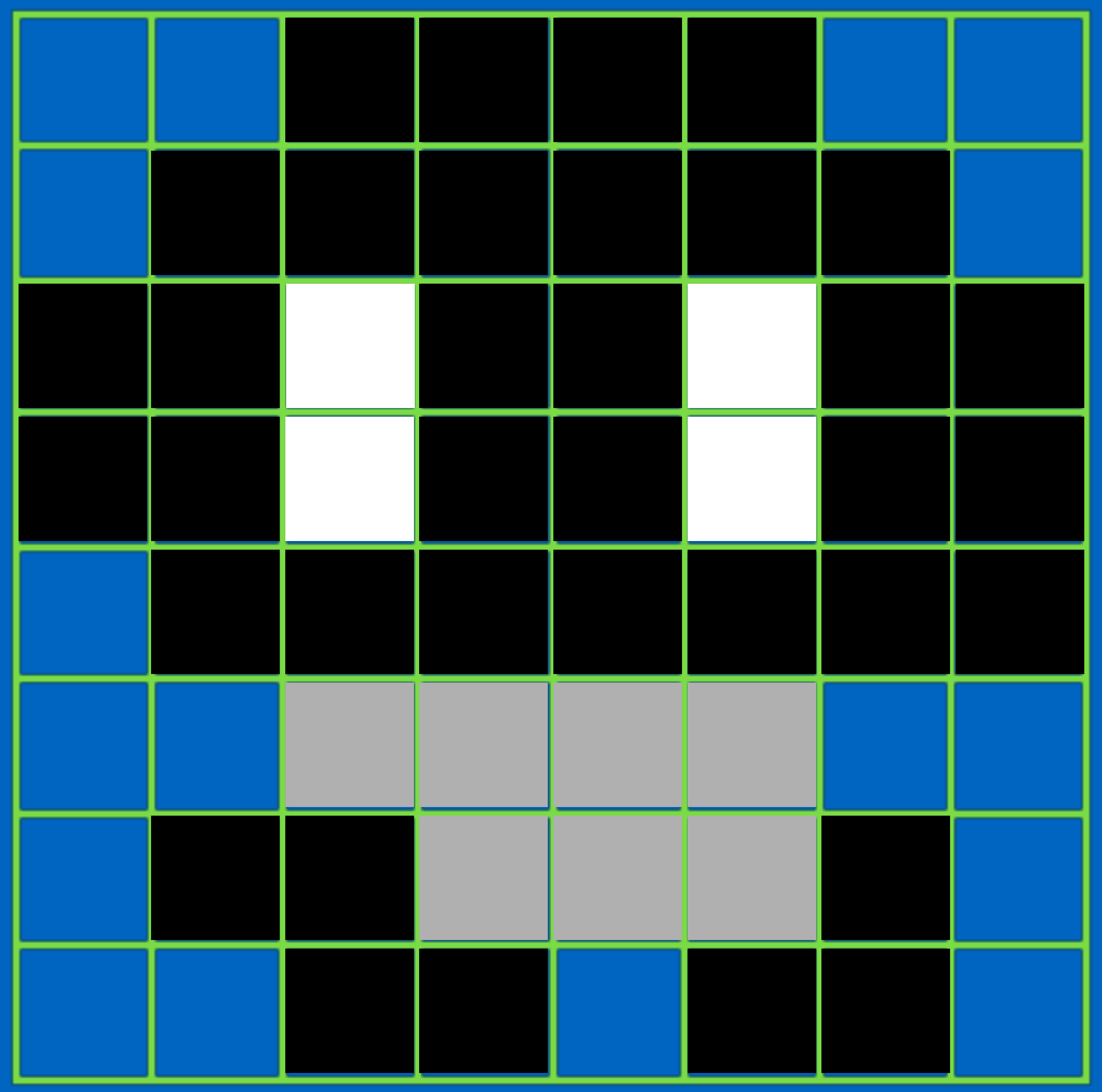


OAM Entry

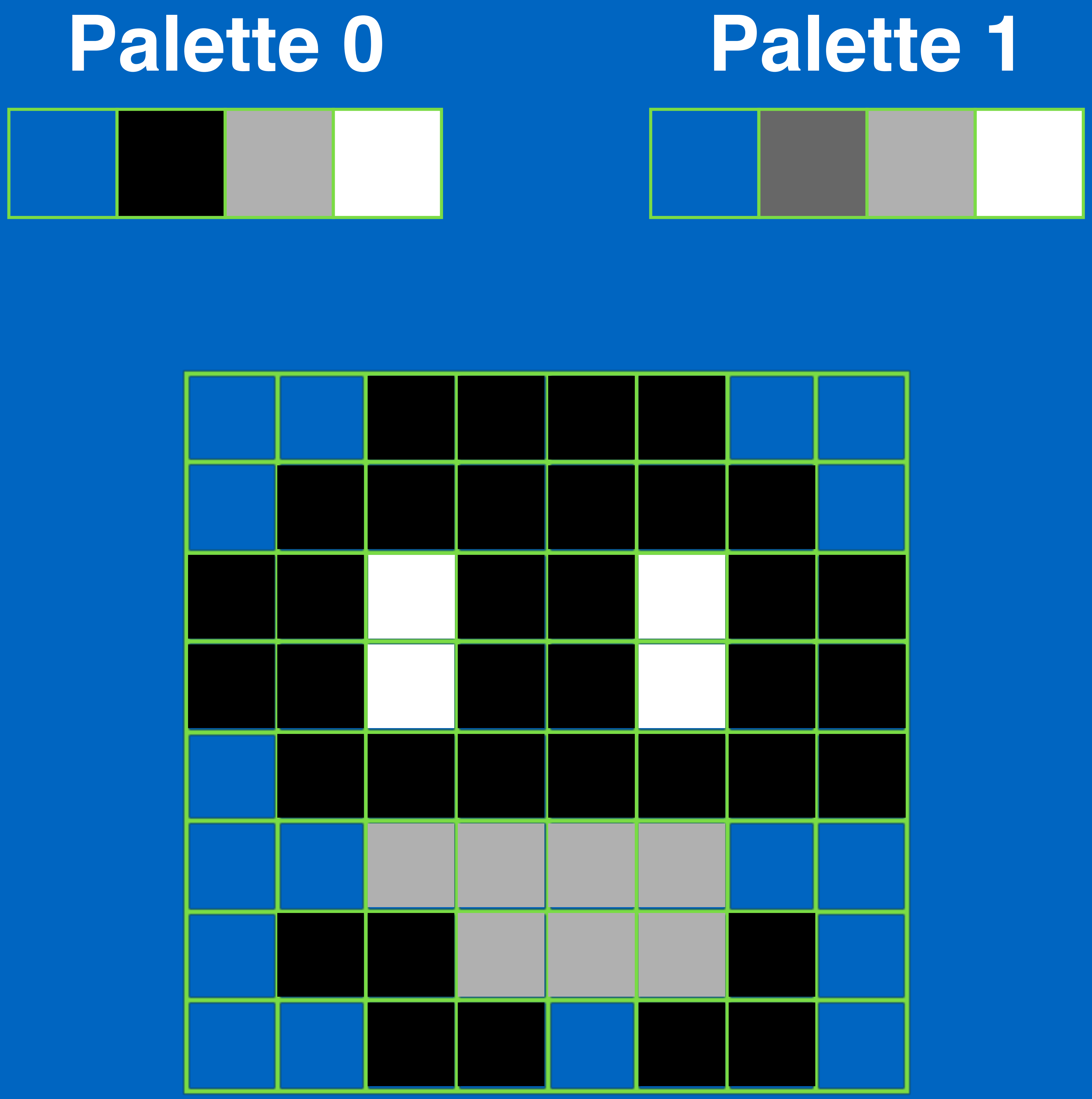
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☒
Palette	_



OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☒
Flip Y	☒
Palette	_



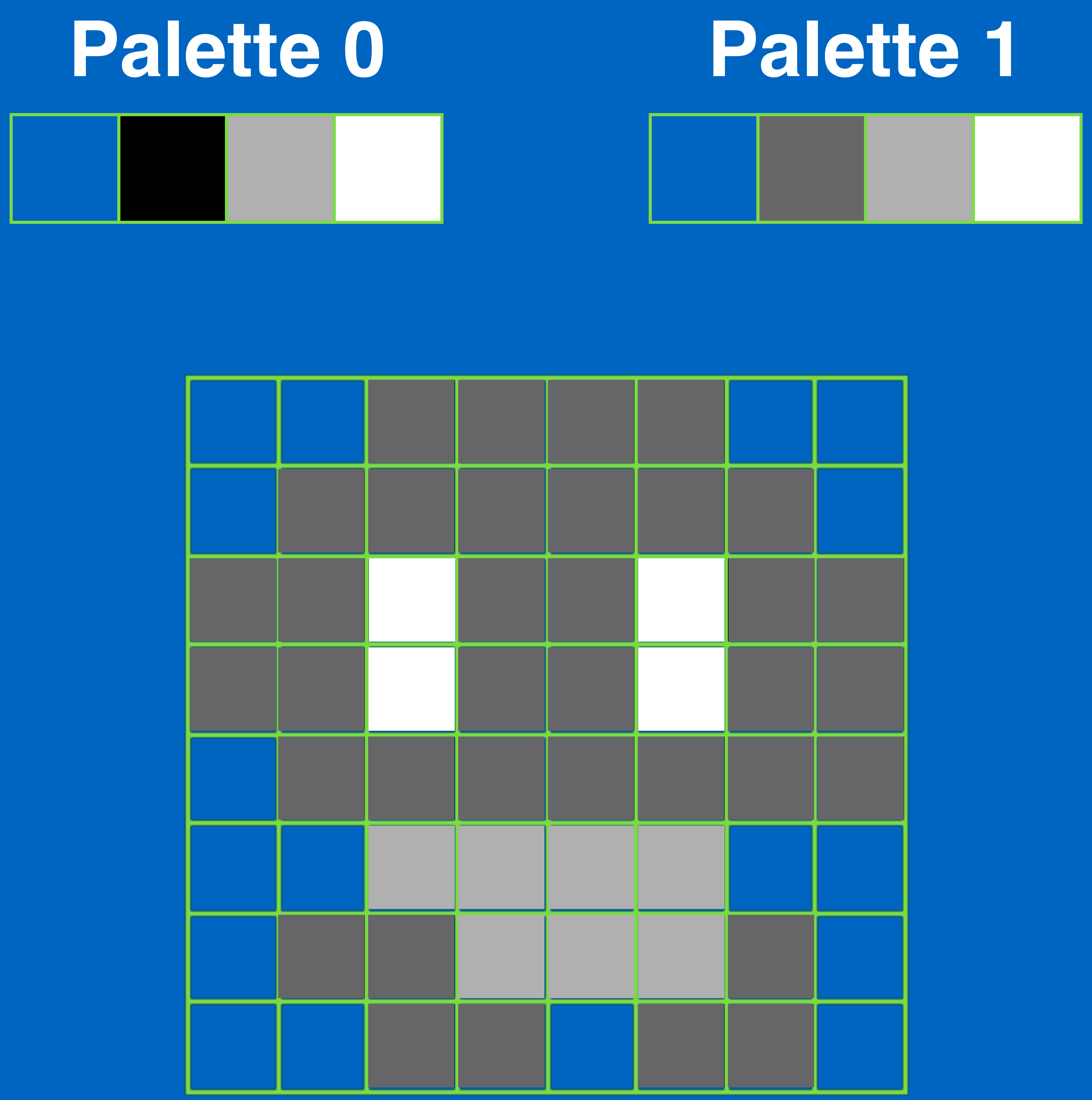
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	_



OAM Entry

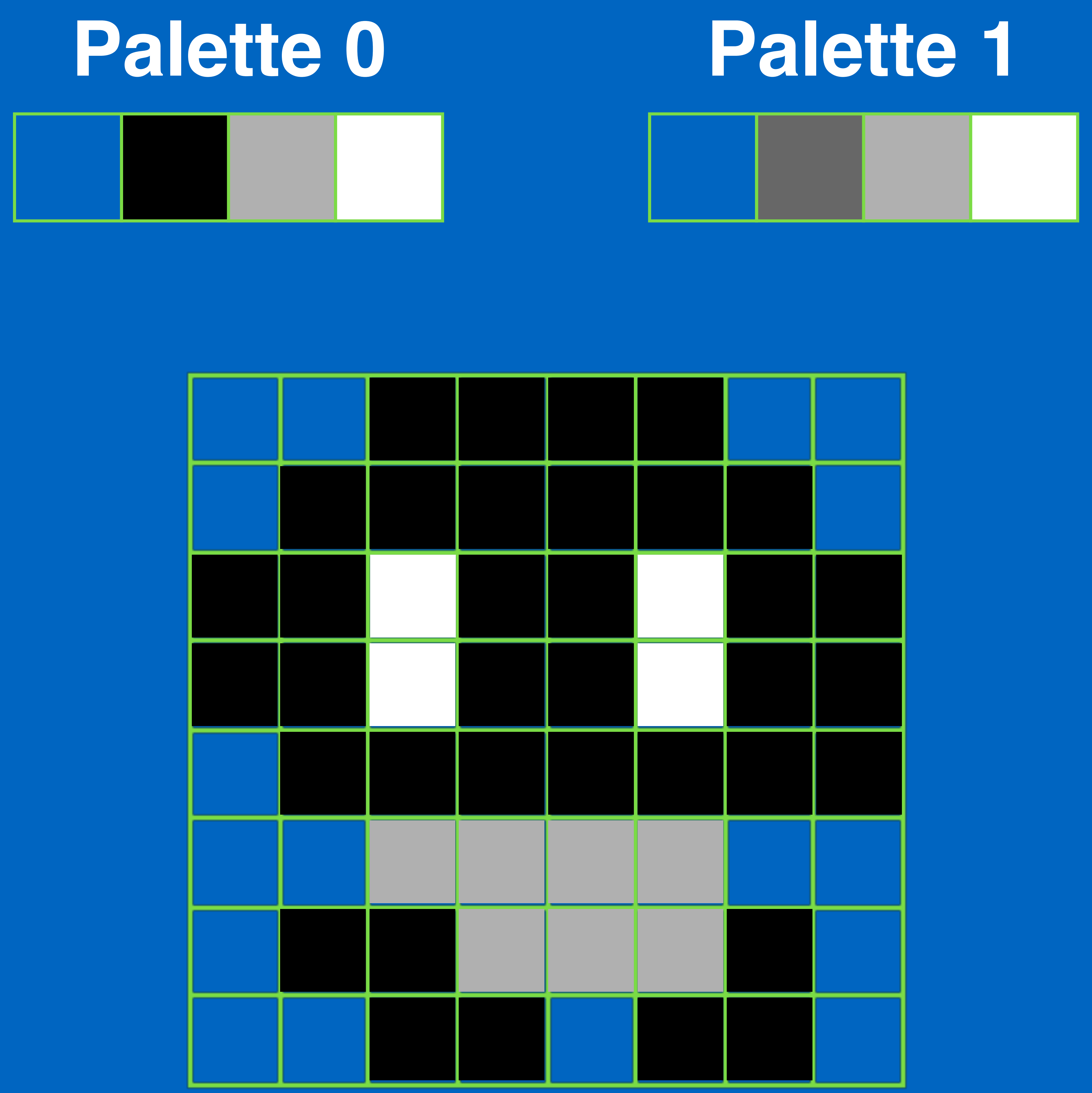
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	_

FF48	OBP0	Object Palette 0 (R/W)
FF49	OBP1	Object Palette 1 (R/W)



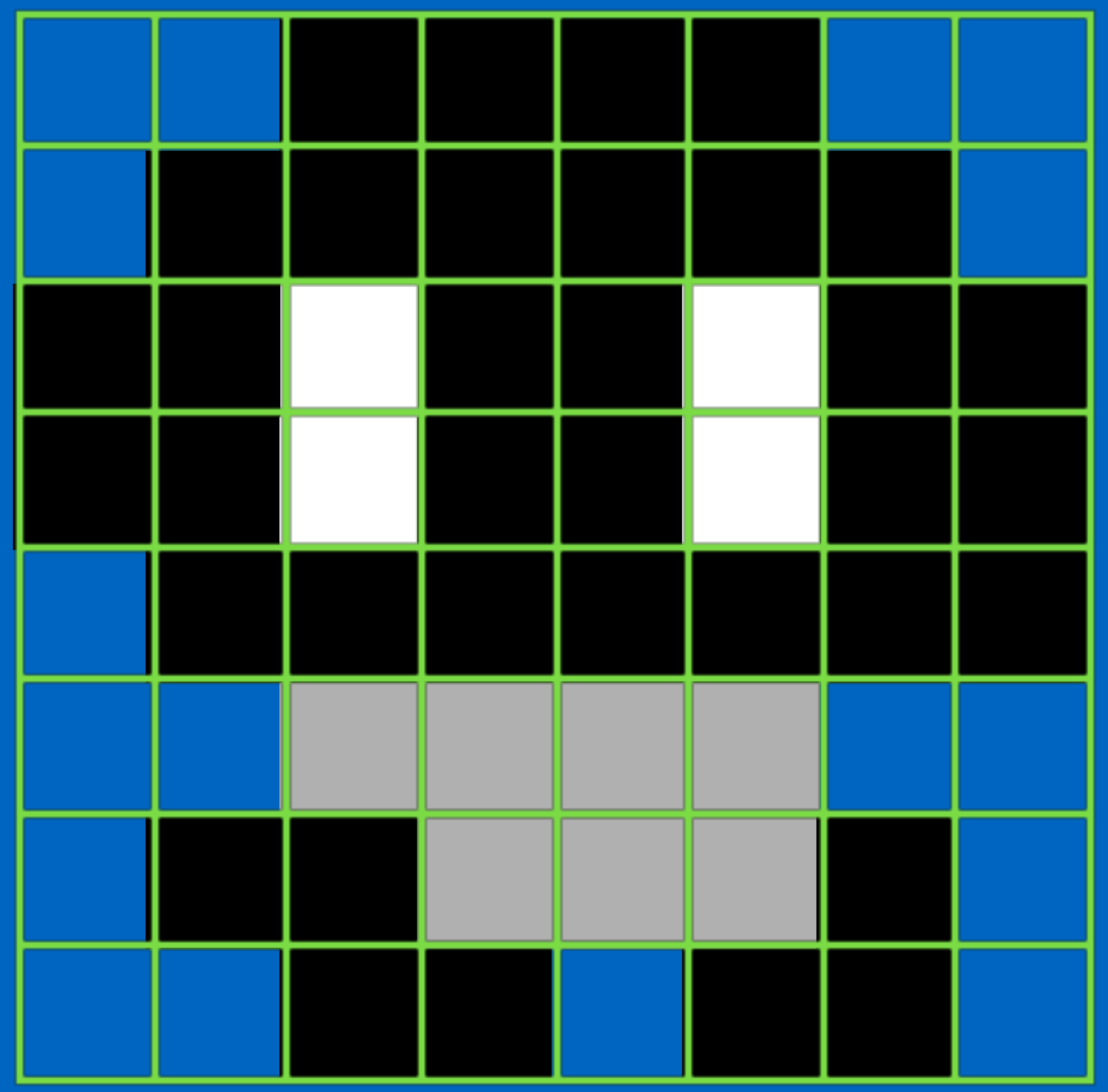
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	1

FF48	OBP0	Object Palette 0 (R/W)
FF49	OBP1	Object Palette 1 (R/W)



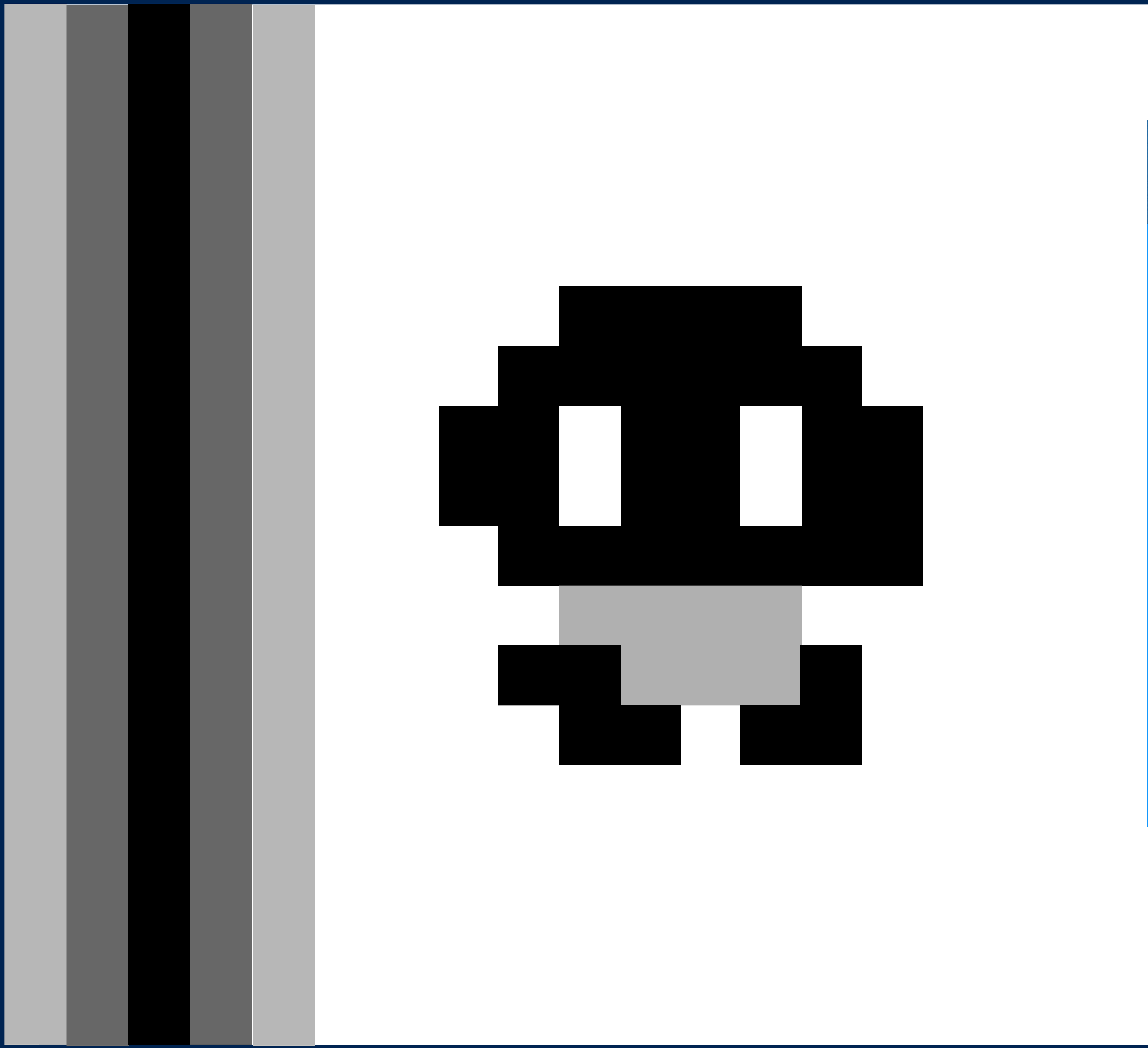
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	0

FF48	OBP0	Object Palette 0 (R/W)
FF49	OBP1	Object Palette 1 (R/W)

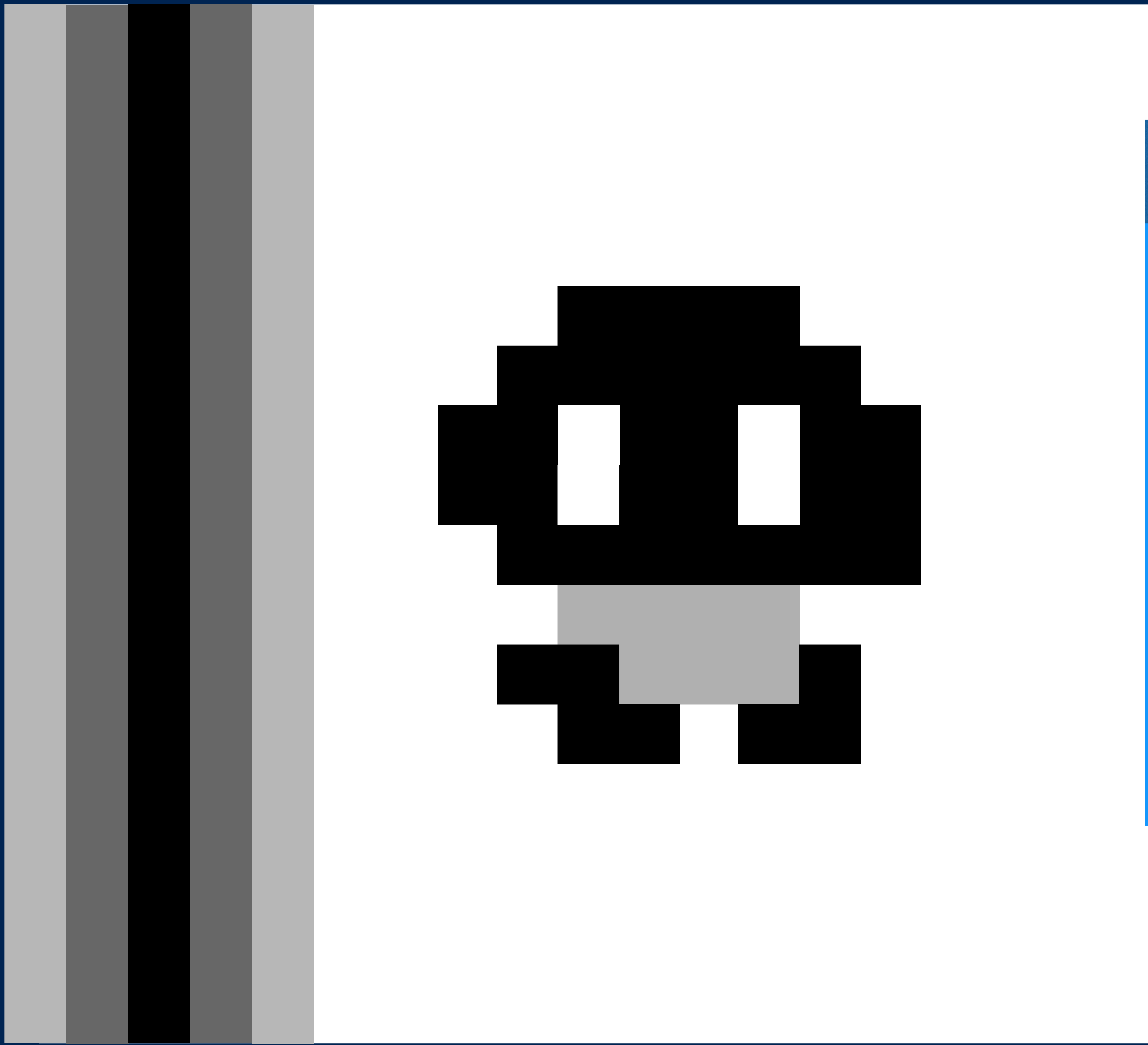


OAM Entry

Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	☐
Flip Y	☐
Palette	0



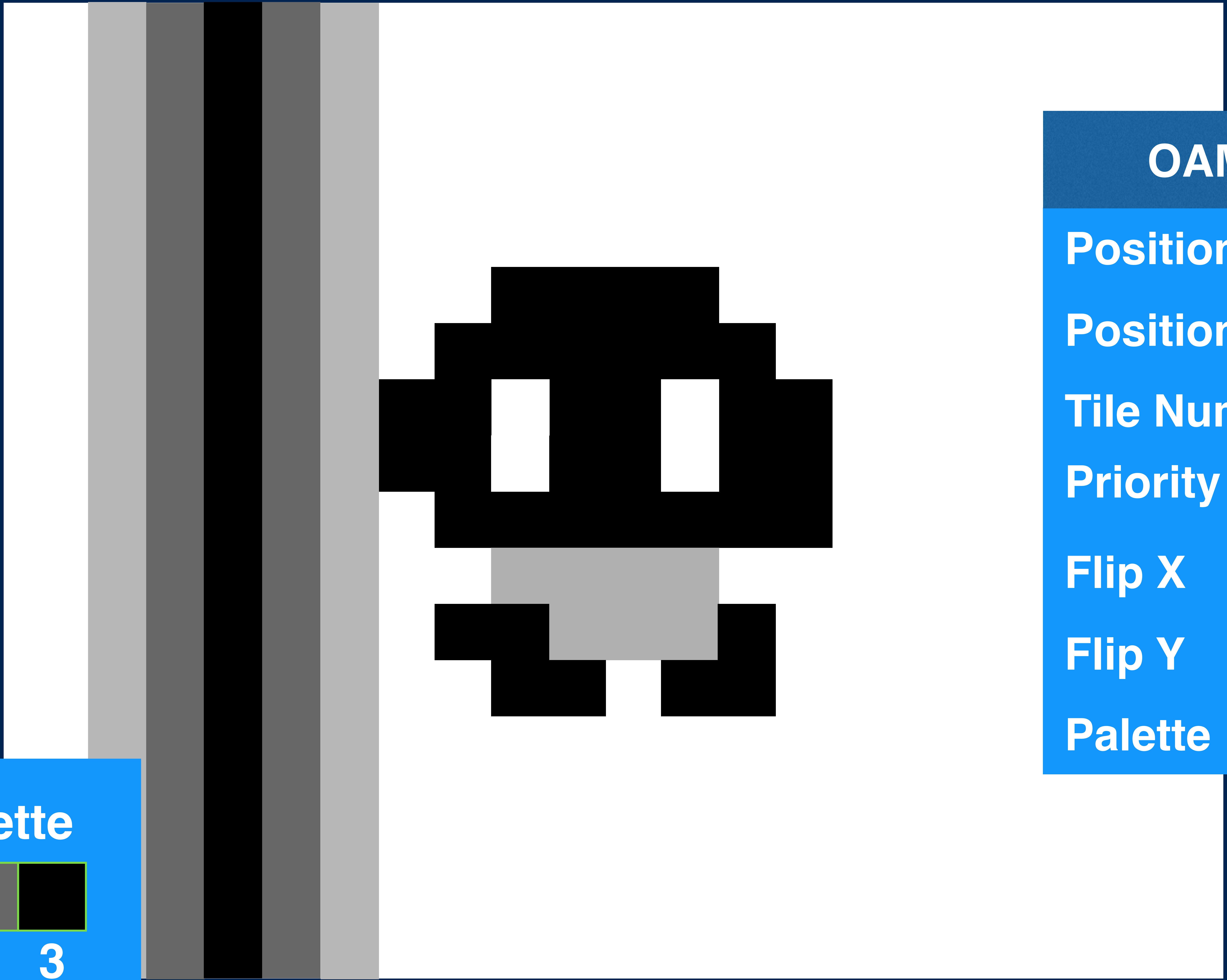
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	_
Flip X	
Flip Y	
Palette	0



OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	1
Flip X	☐
Flip Y	☐
Palette	0


OAM Entry
Position X 0x4D
Position Y 0x78
Tile Number 0x90
Priority 1
Flip X ☐
Flip Y ☐
Palette 0
BG Palette
0 1 2 3


OAM Entry
Position X 0x4D
Position Y 0x78
Tile Number 0x90
Priority 1
Flip X
Flip Y
Palette 0
BG Palette
0 1 2 3



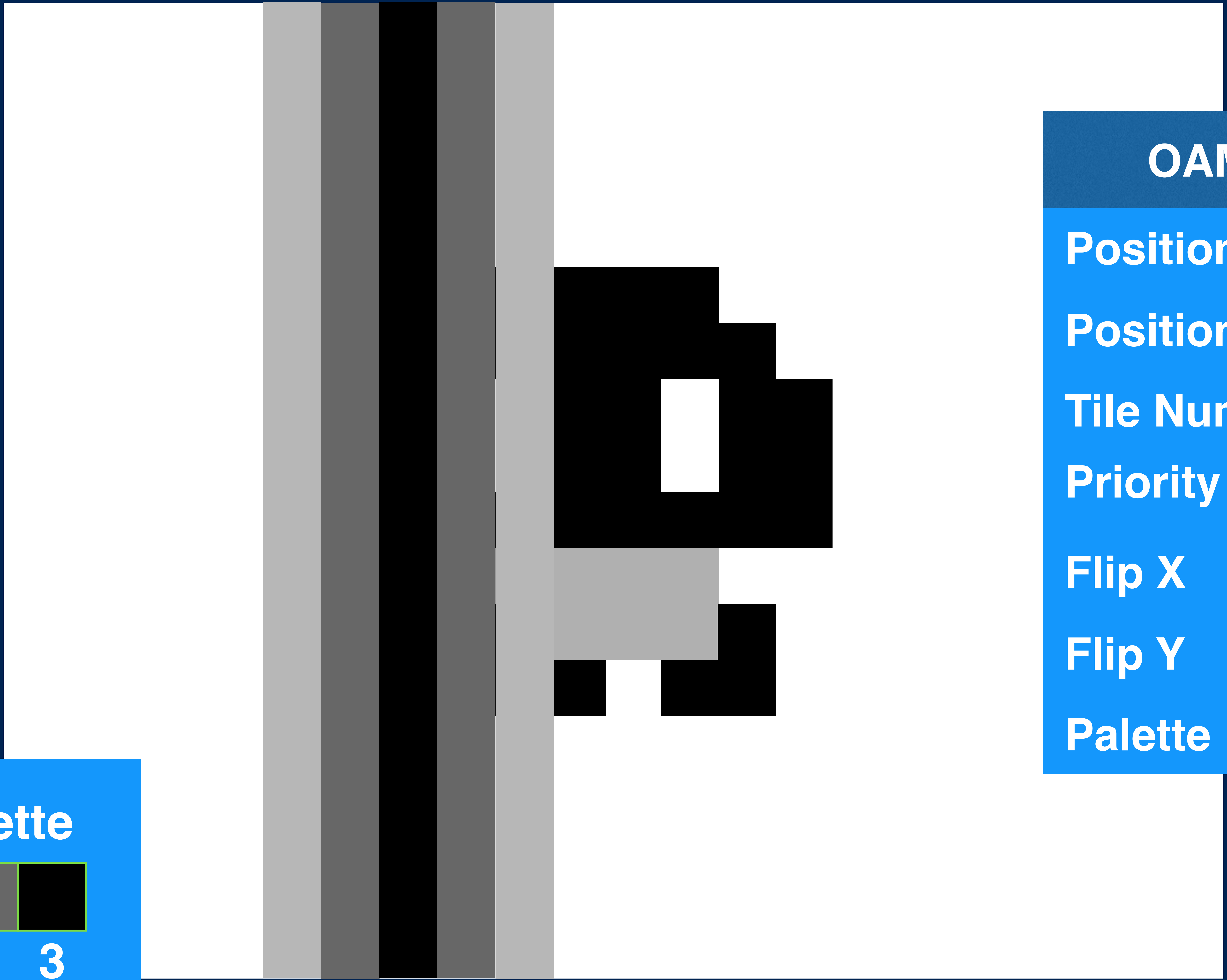
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	1
Flip X	☐
Flip Y	☐
Palette	0

BG Palette

0 1 2 3

OAM Entry
Position X 0x4D
Position Y 0x78
Tile Number 0x90
Priority 1
Flip X ☐
Flip Y ☐
Palette 0
BG Palette
0 1 2 3


OAM Entry
Position X 0x4D
Position Y 0x78
Tile Number 0x90
Priority 1
Flip X
Flip Y
Palette 0
BG Palette
0 1 2 3



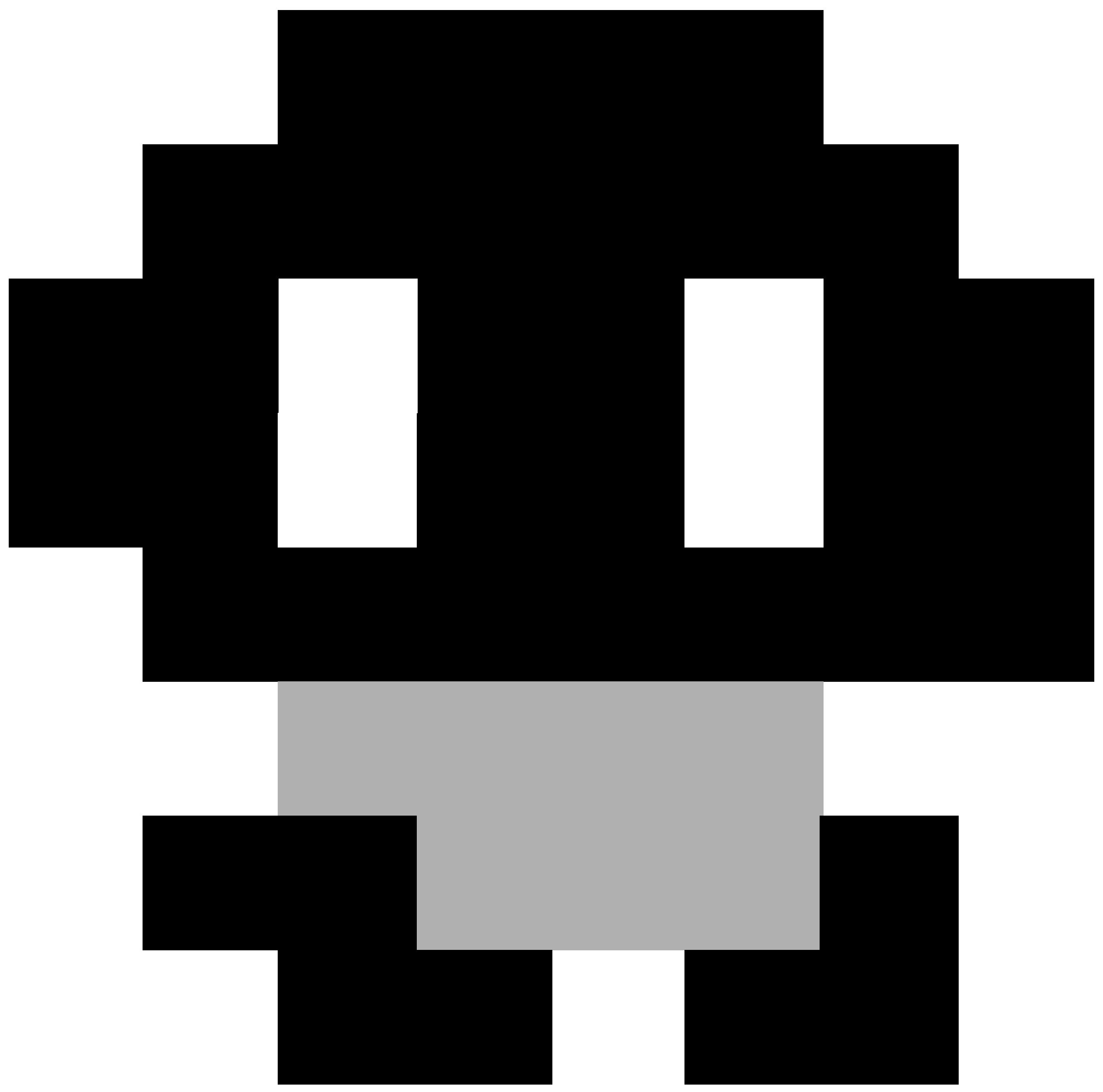
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	1
Flip X	☐
Flip Y	☐
Palette	0

BG Palette

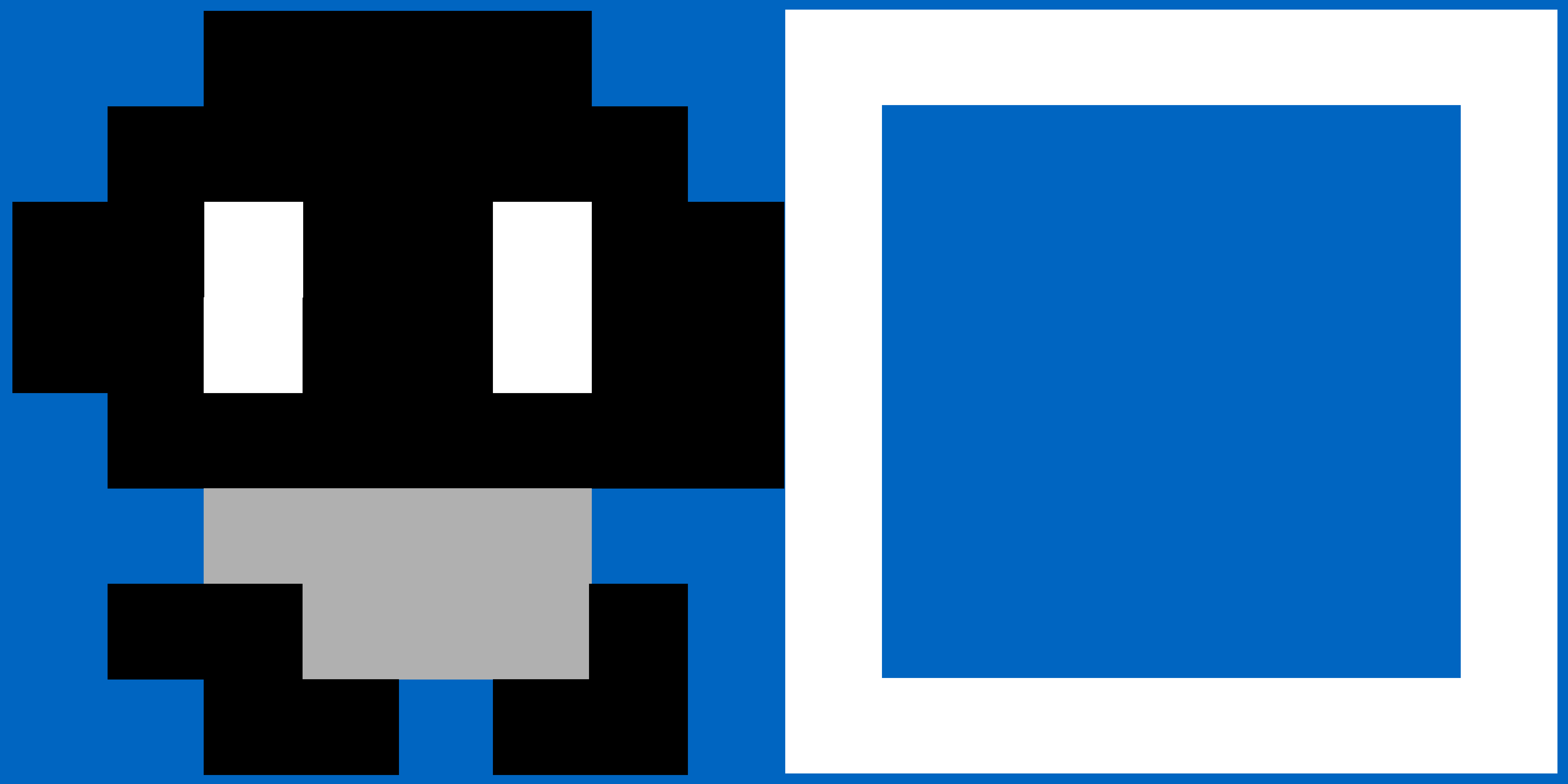
0 1 2 3


OAM Entry
Position X 0x4D
Position Y 0x78
Tile Number 0x90
Priority 1
Flip X ☐
Flip Y ☐
Palette 0
BG Palette
0 1 2 3

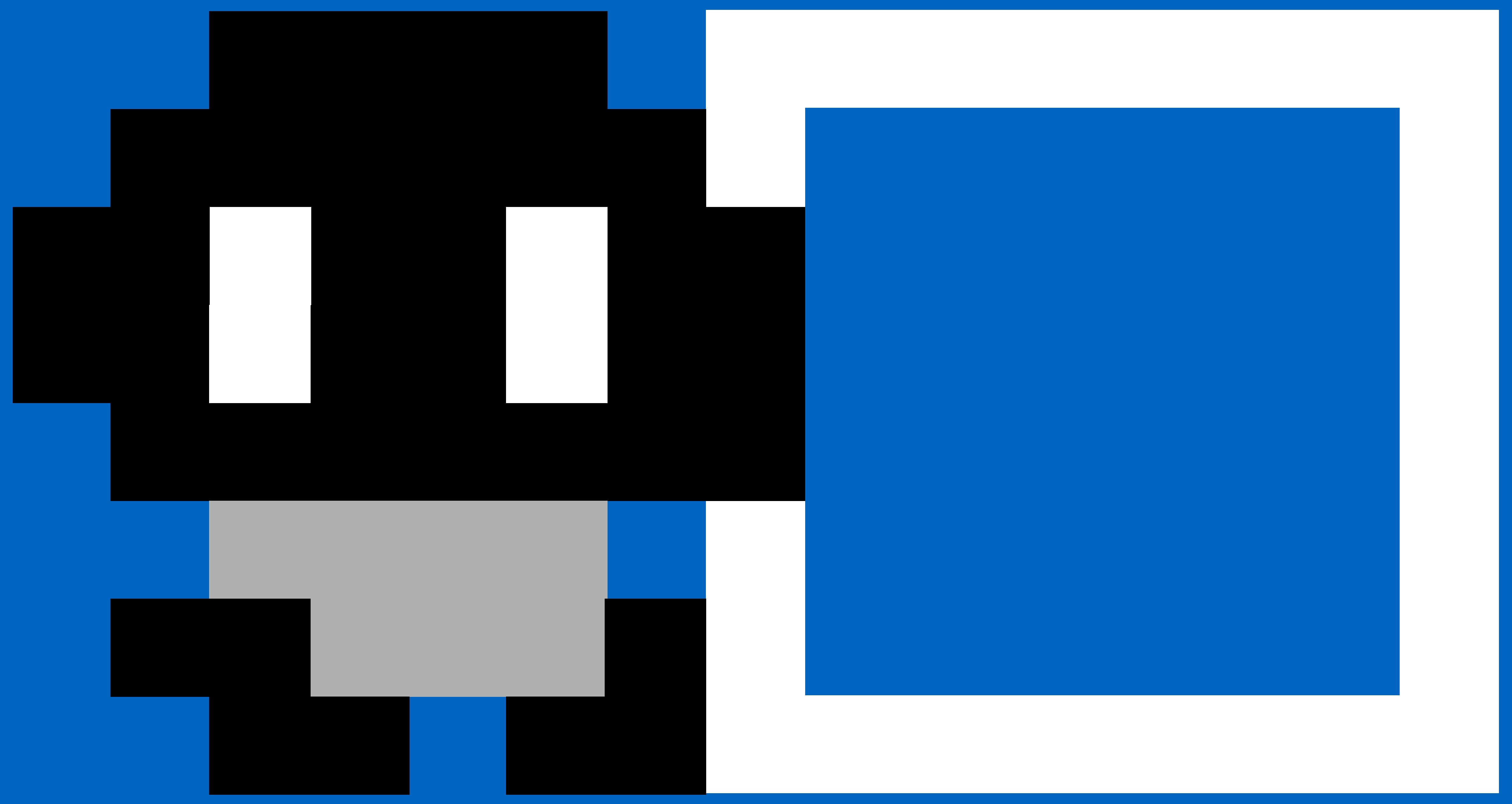

OAM Entry
Position X 0x4D
Position Y 0x78
Tile Number 0x90
Priority 0
Flip X ☐
Flip Y ☐
Palette 0
BG Palette
0 1 2 3



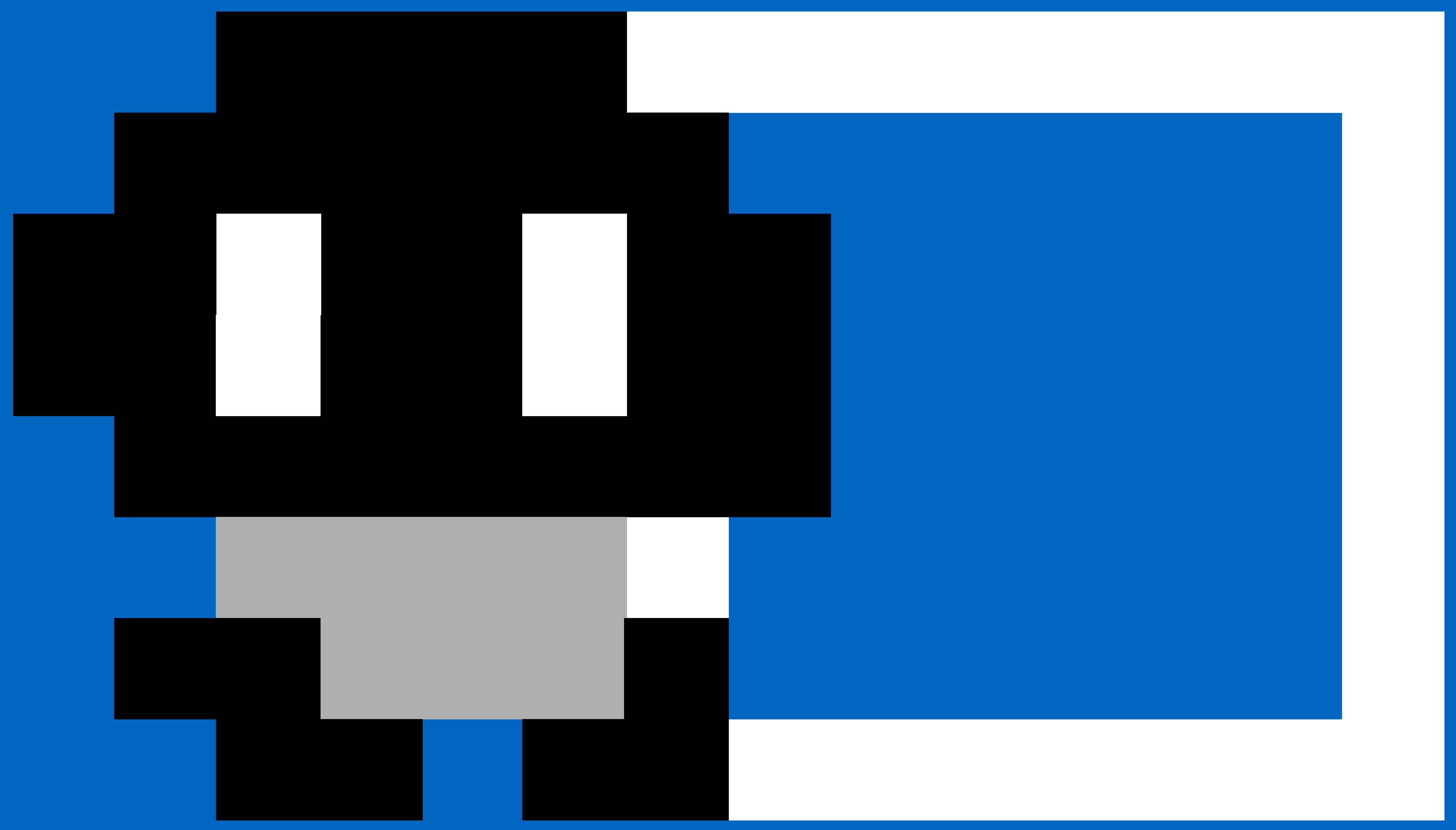
Sprite/Sprite Priority



Sprite/Sprite Priority

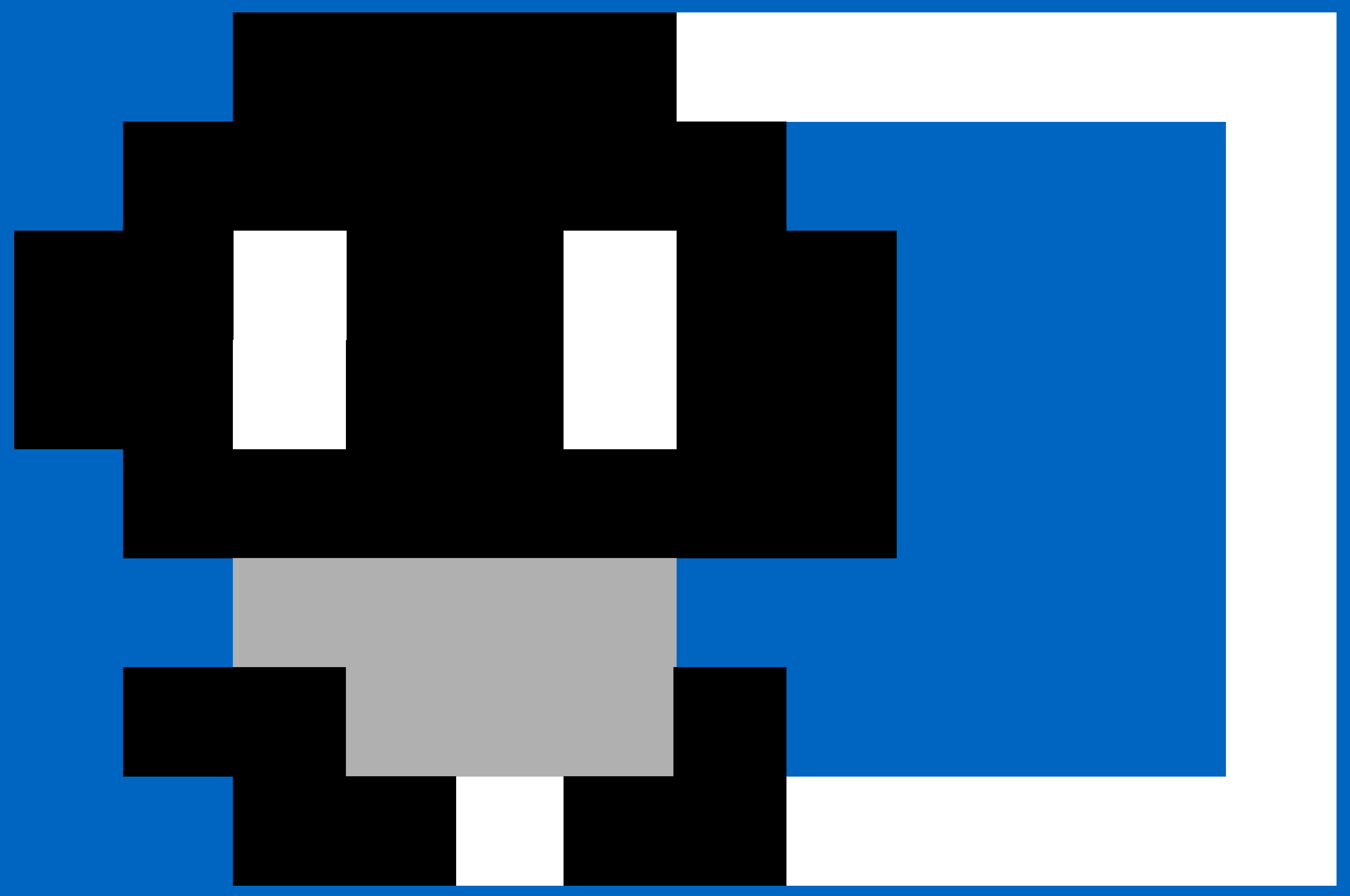


Sprite/Sprite Priority

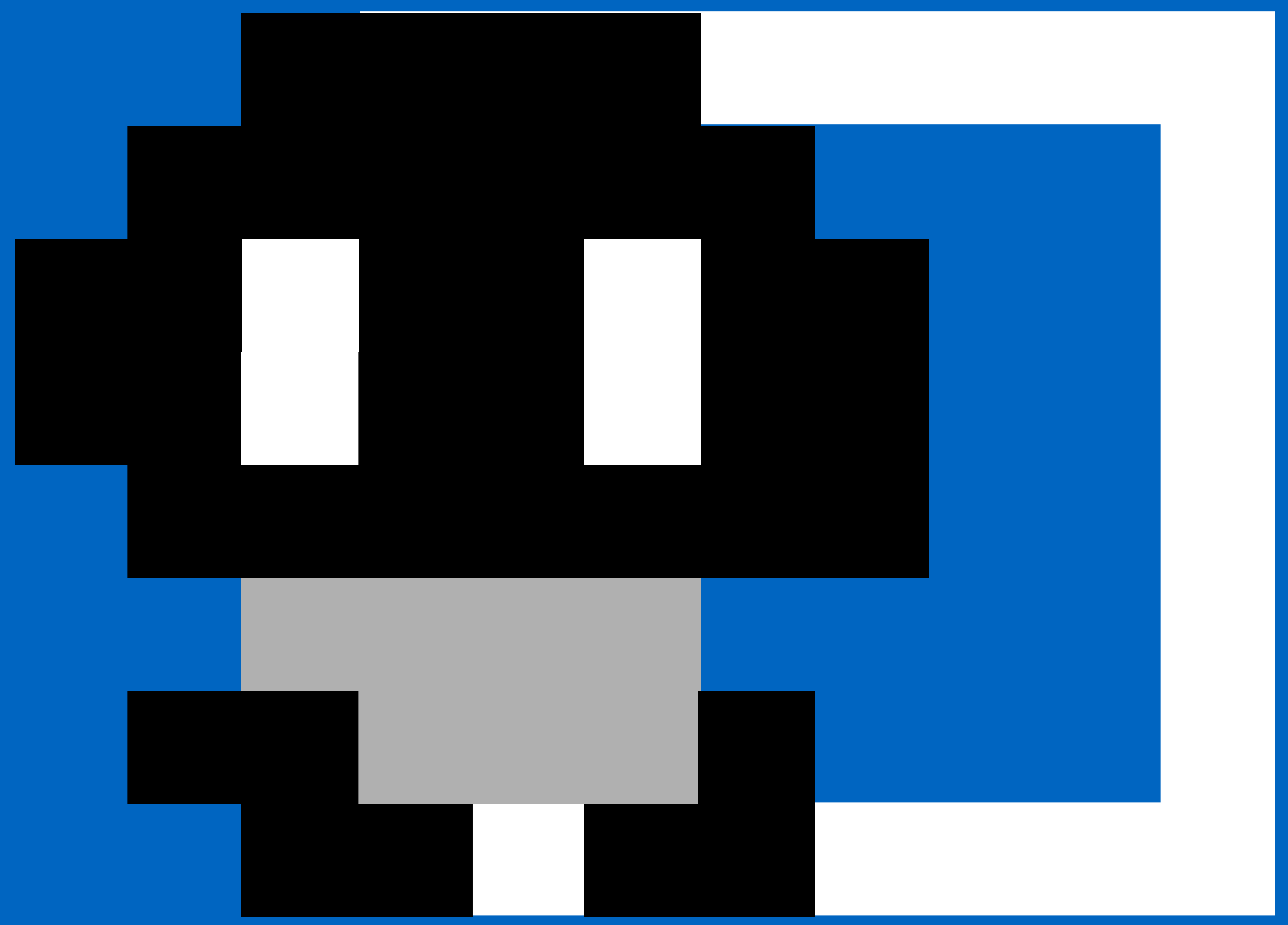


Sprite/Sprite Priority

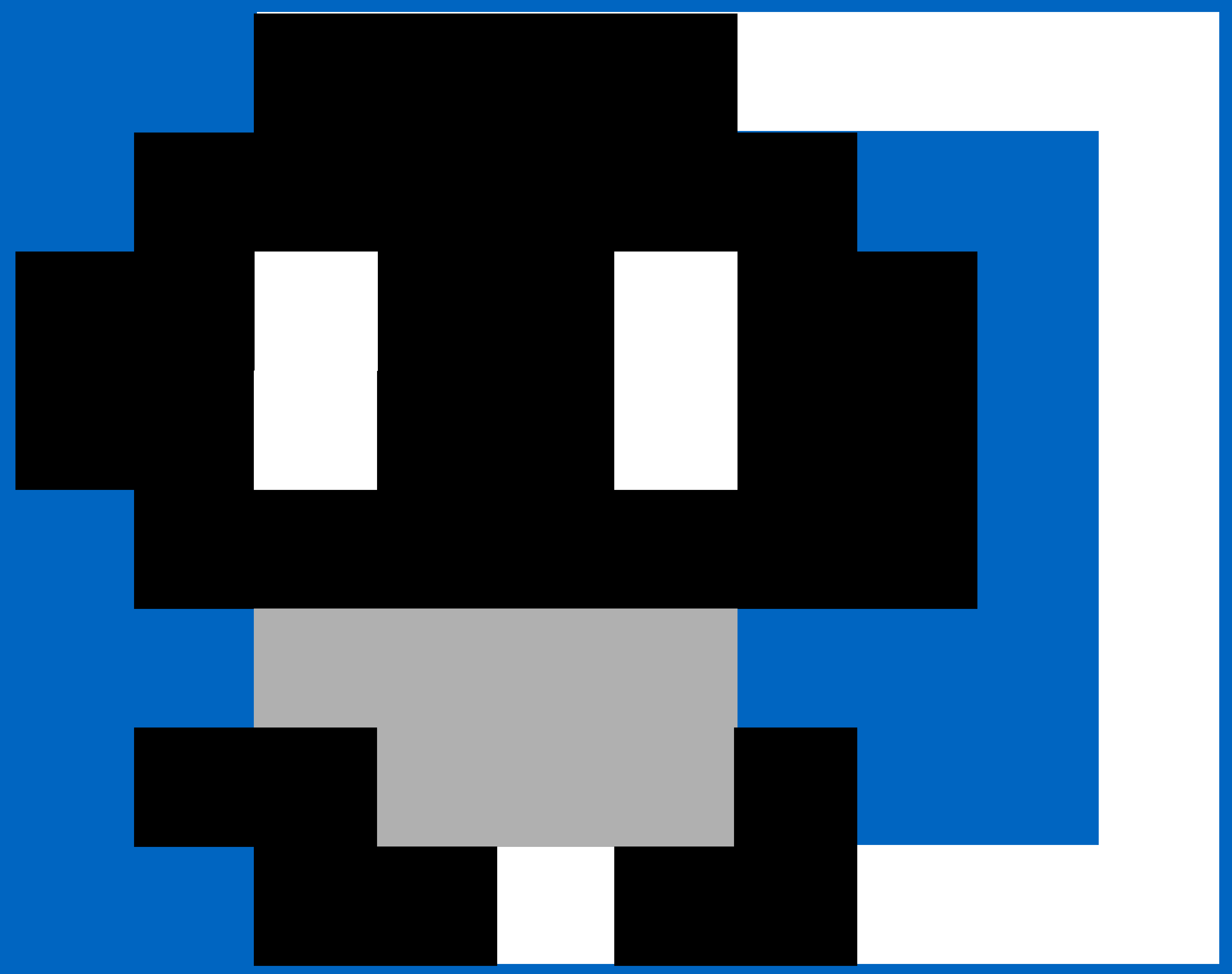
Sprite/Sprite Priority



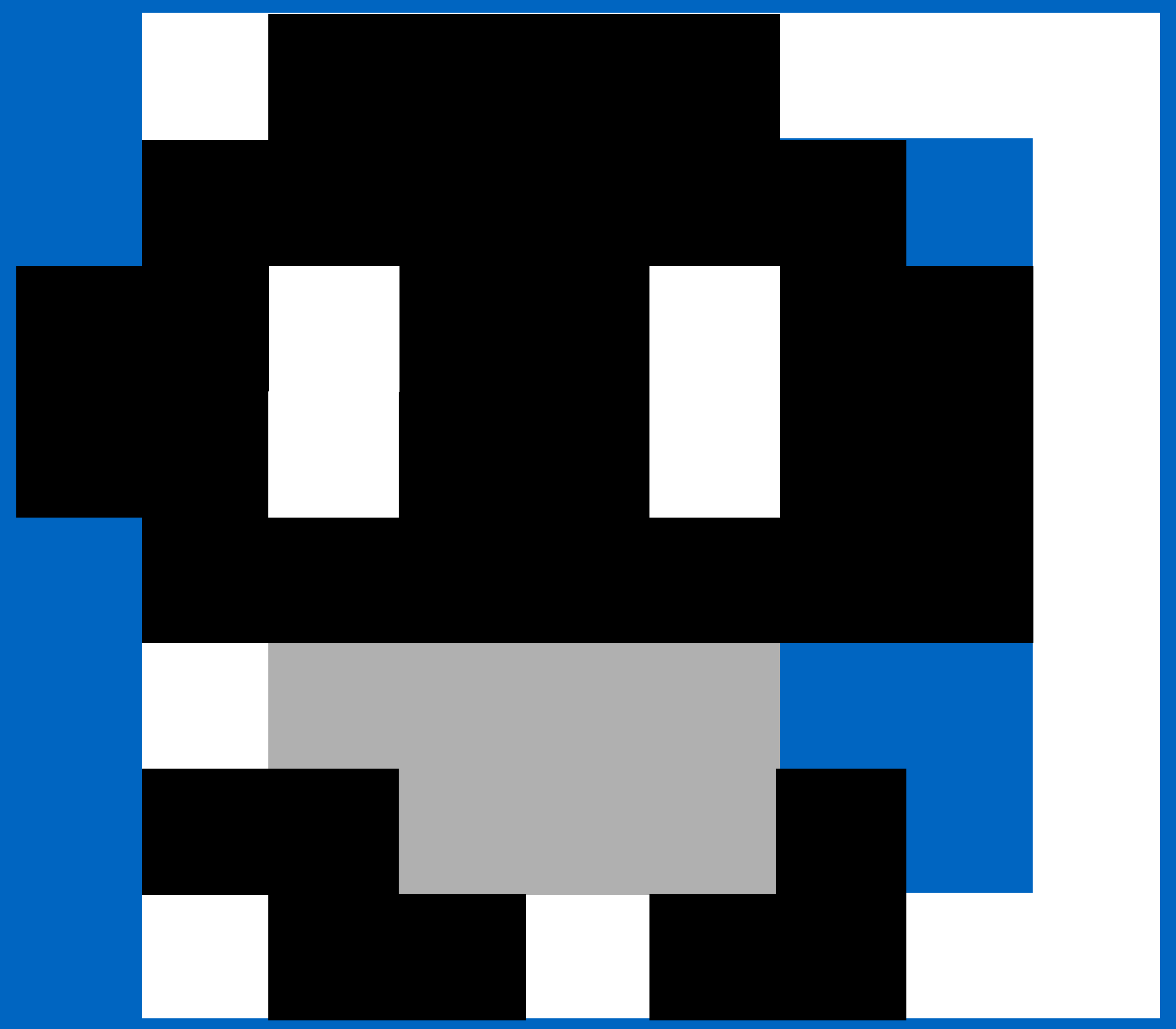
Sprite/Sprite Priority



Sprite/Sprite Priority

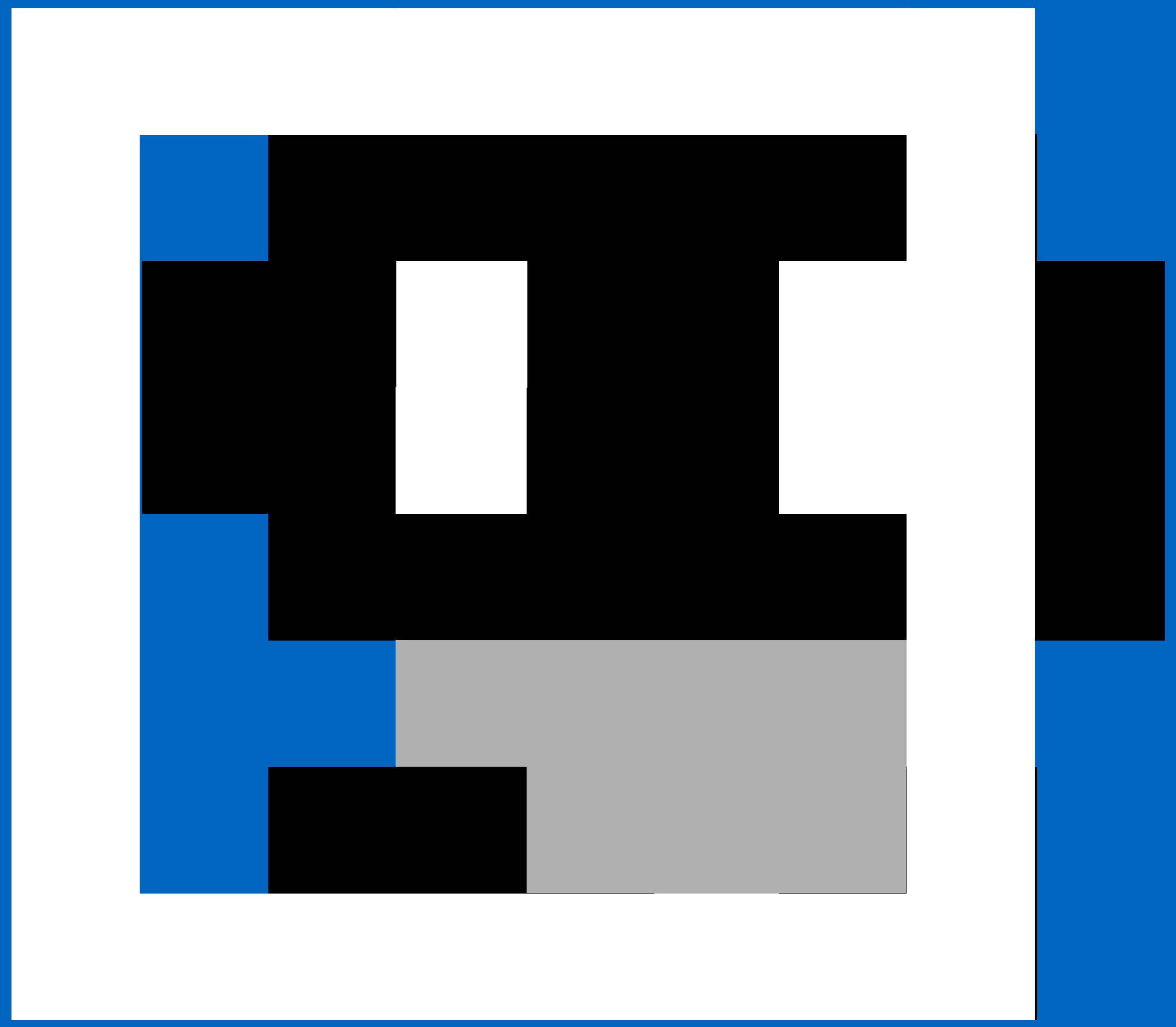


Sprite/Sprite Priority

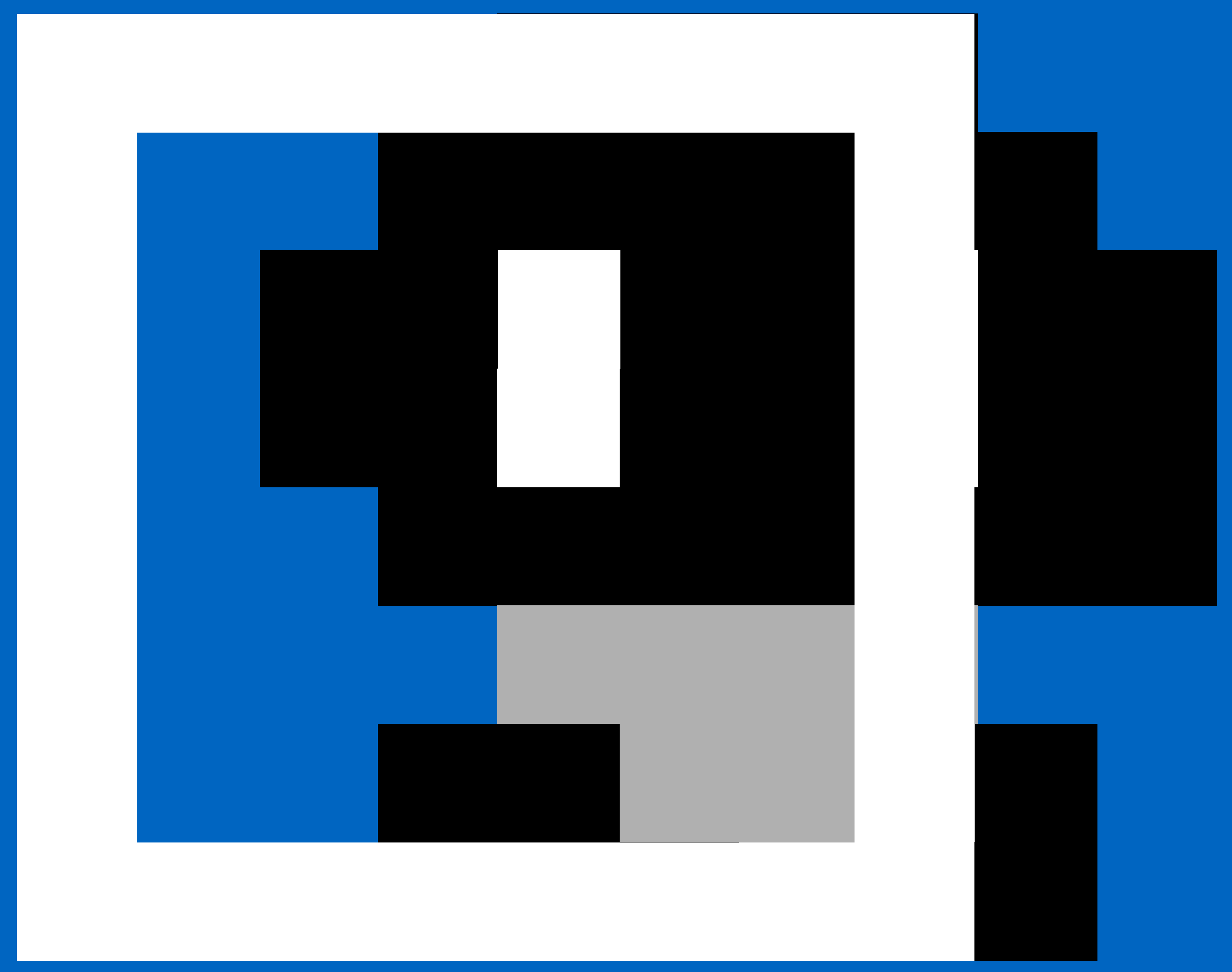


Sprite/Sprite Priority

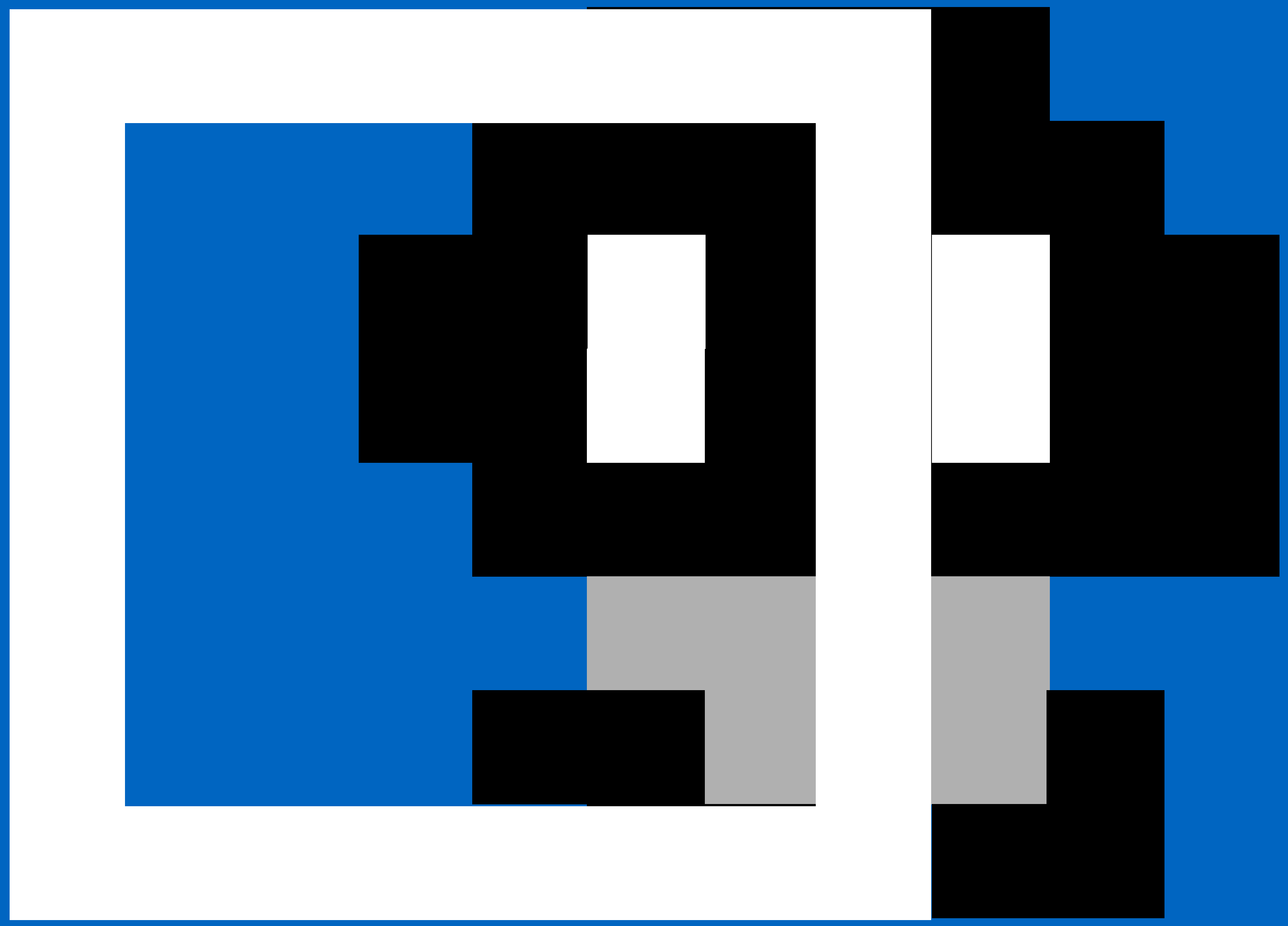
Sprite/Sprite Priority



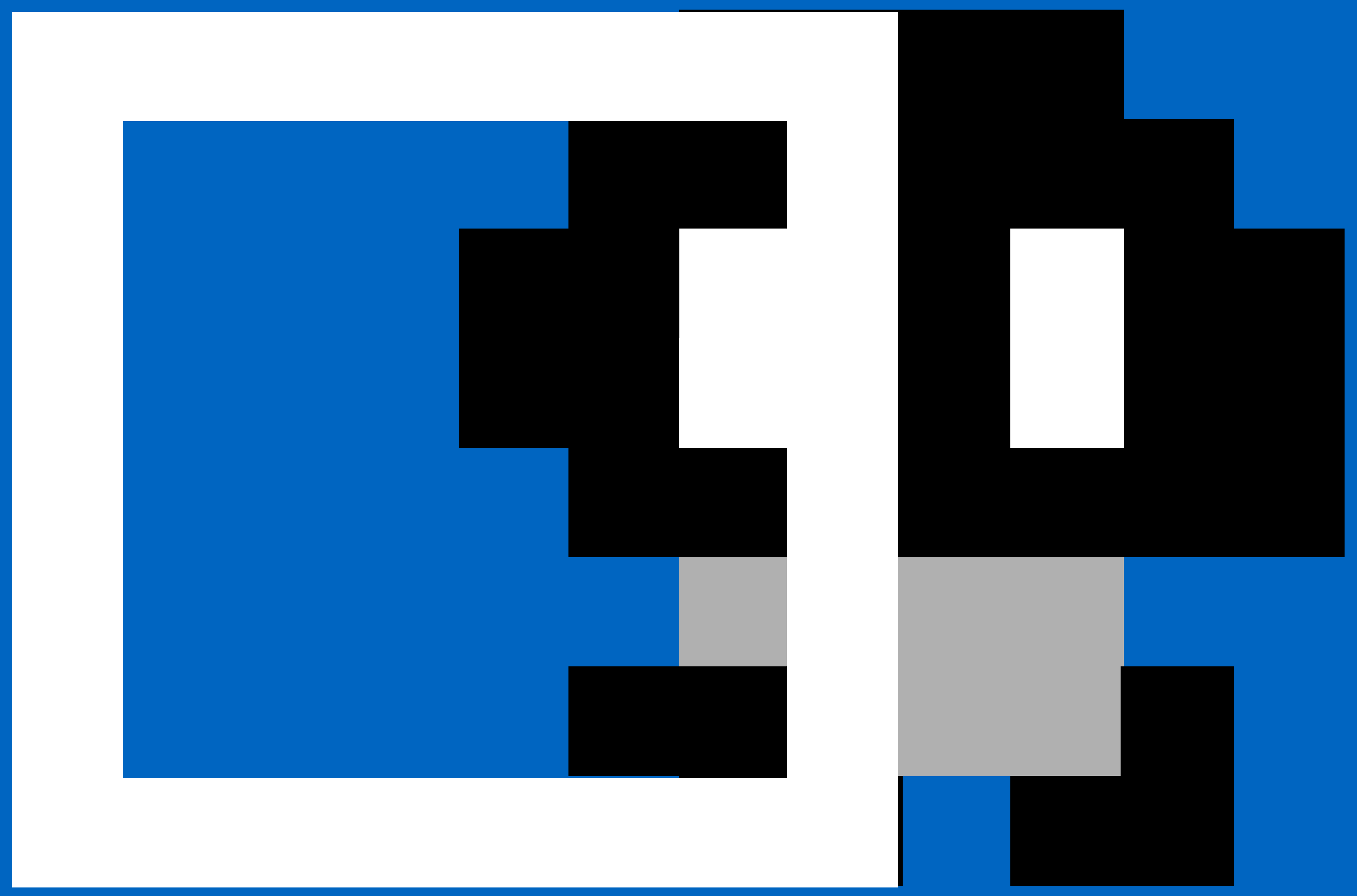
Sprite/Sprite Priority



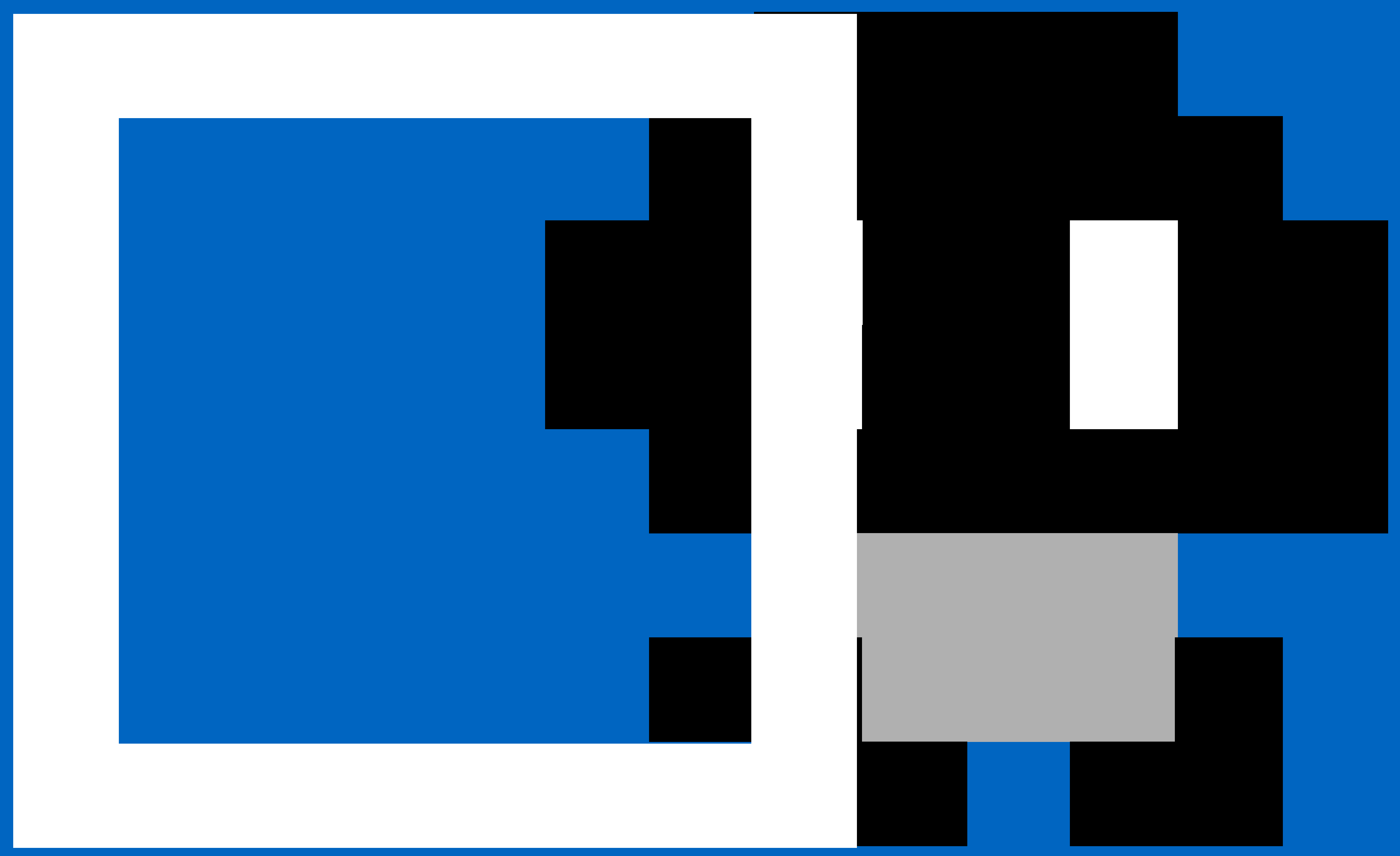
Sprite/Sprite Priority



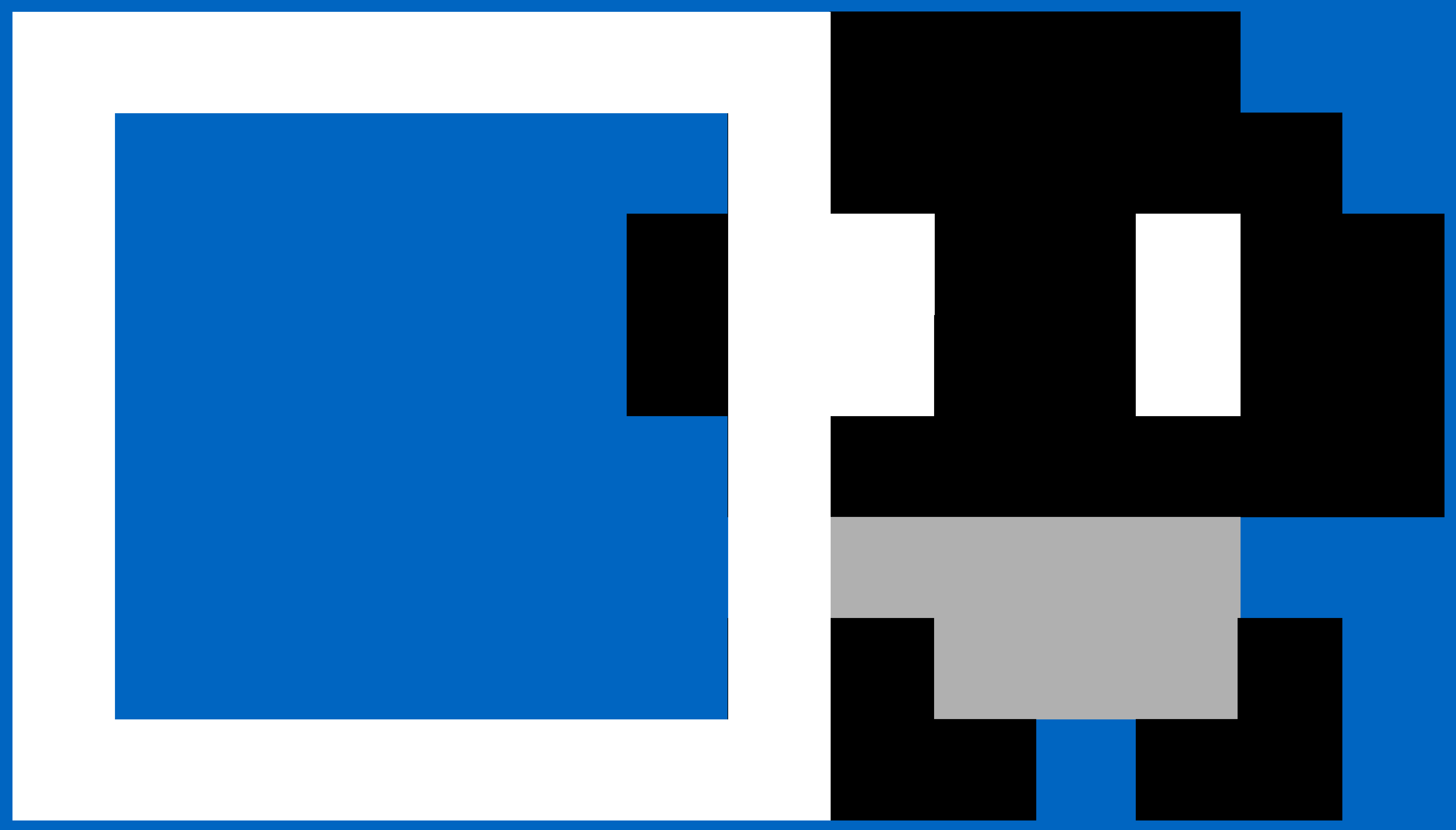
Sprite/Sprite Priority



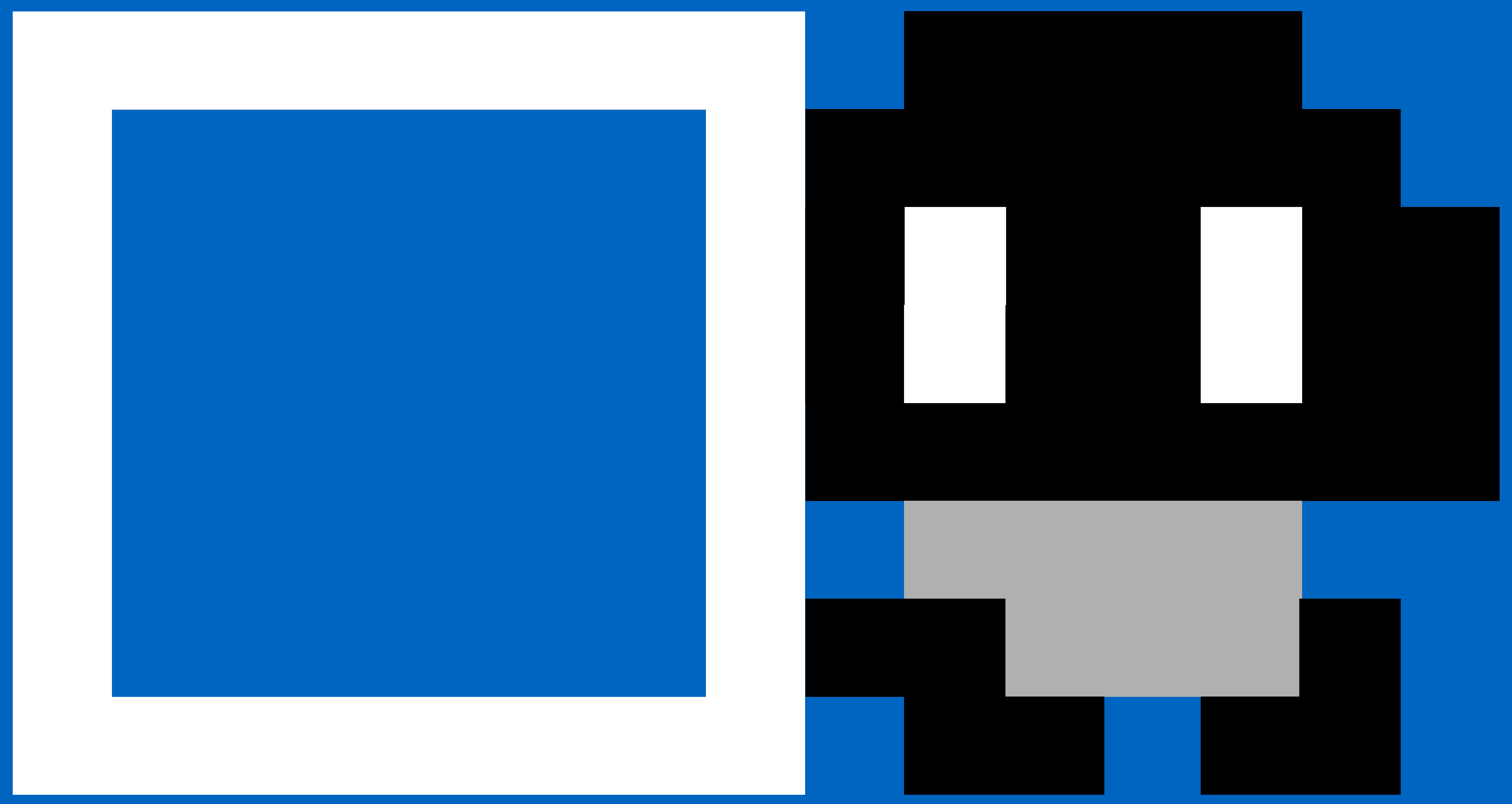
Sprite/Sprite Priority



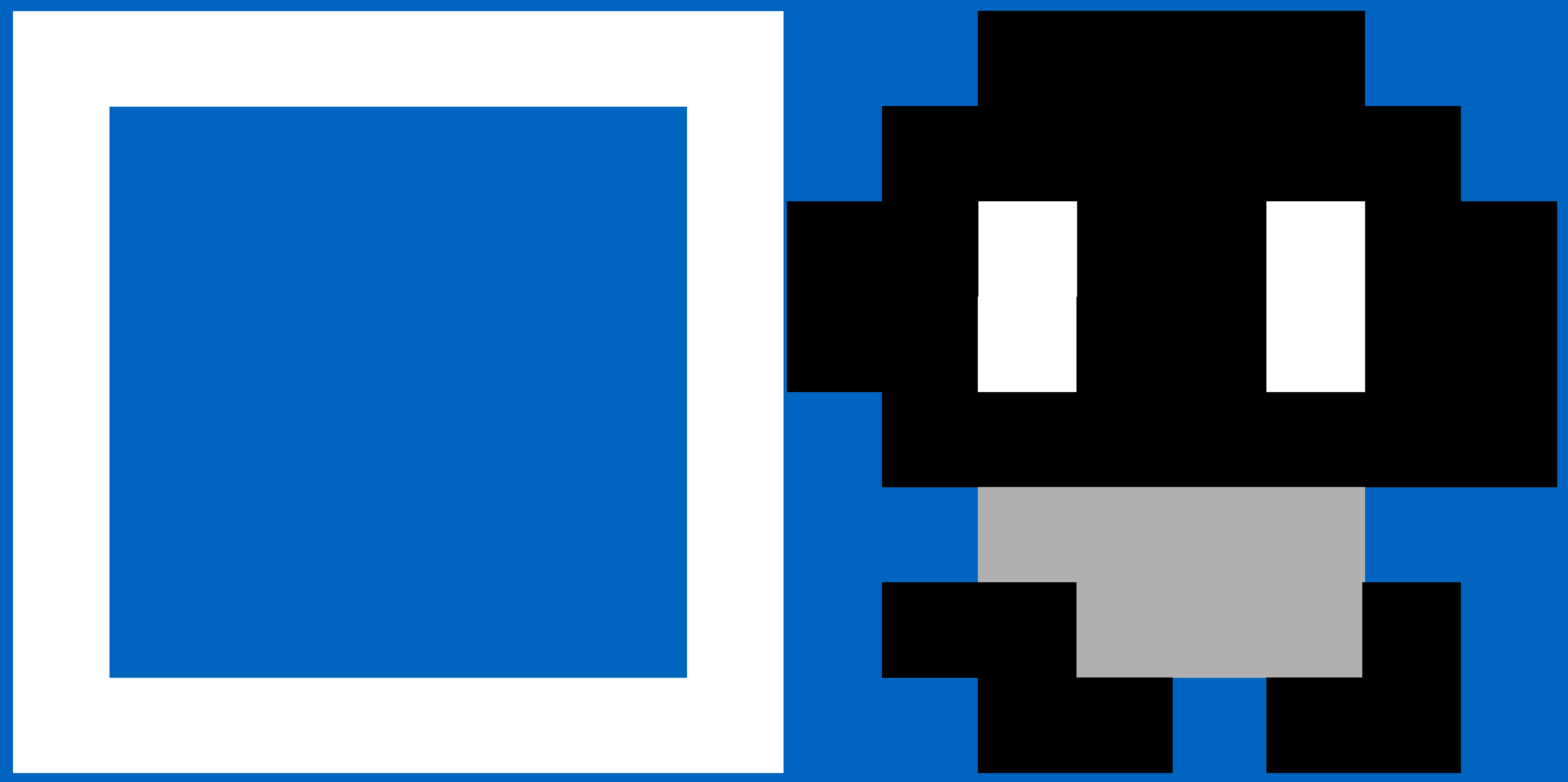
Sprite/Sprite Priority



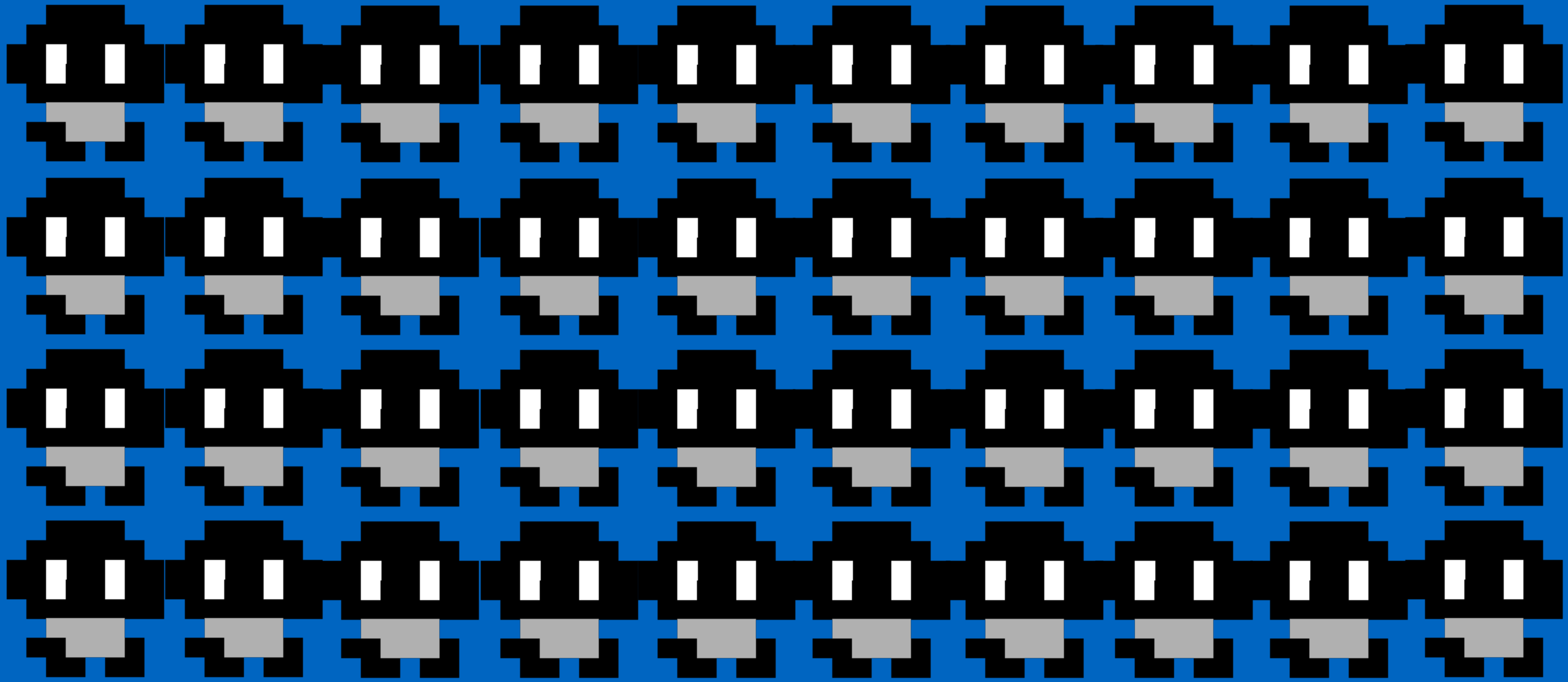
Sprite/Sprite Priority

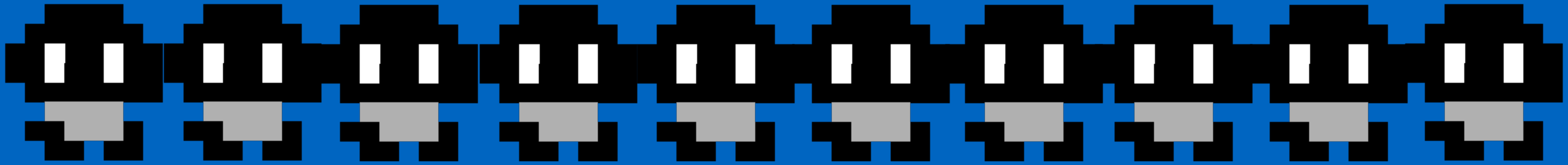


Sprite/Sprite Priority

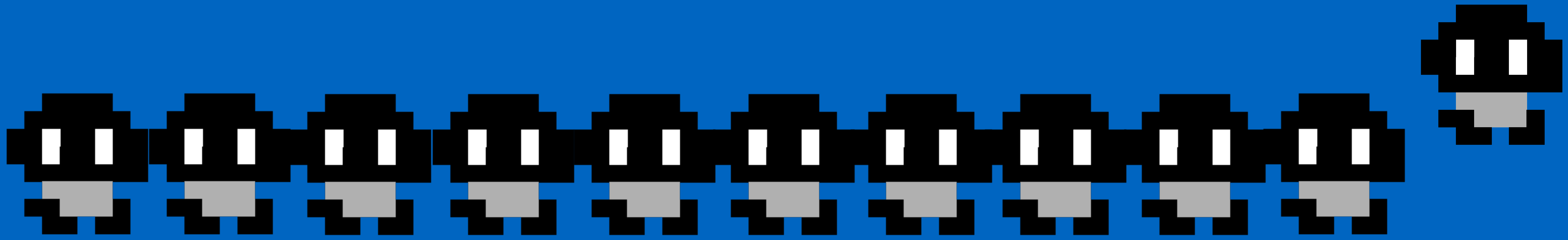


40 sprites total





10 sprites per line



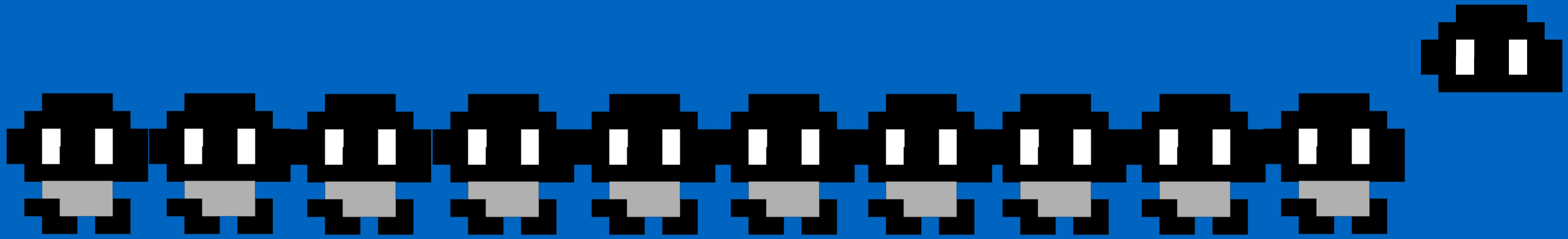
10 sprites per line

10 sprites per line

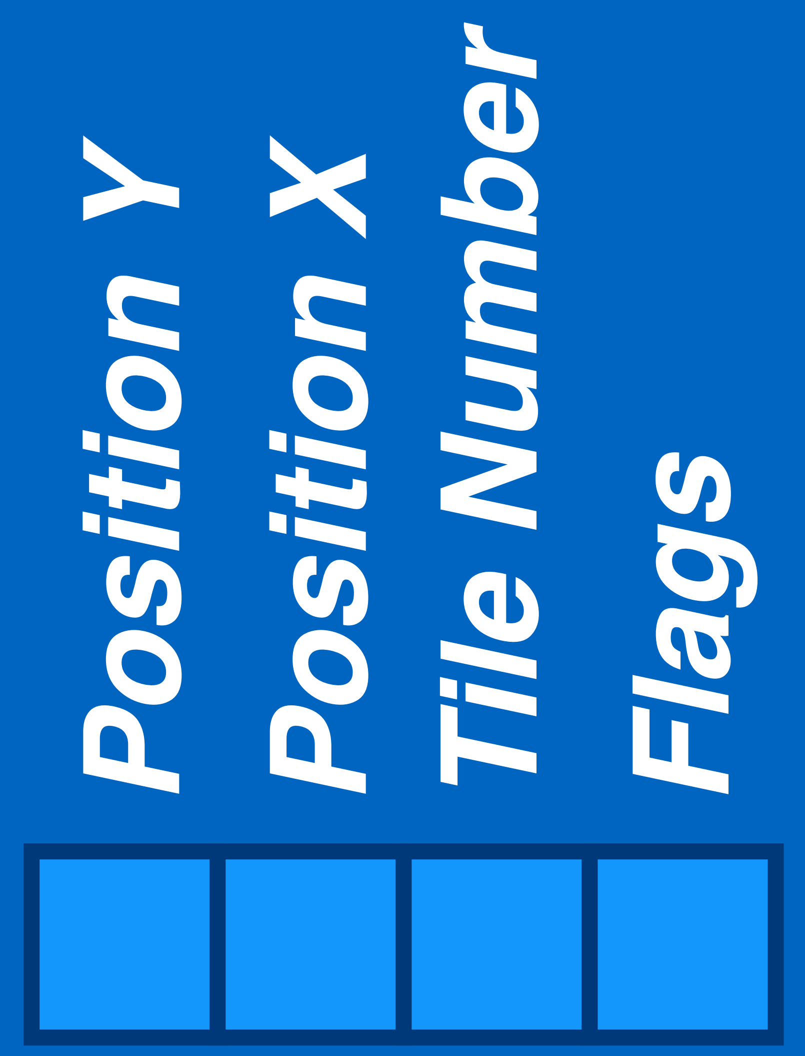
10 sprites per line

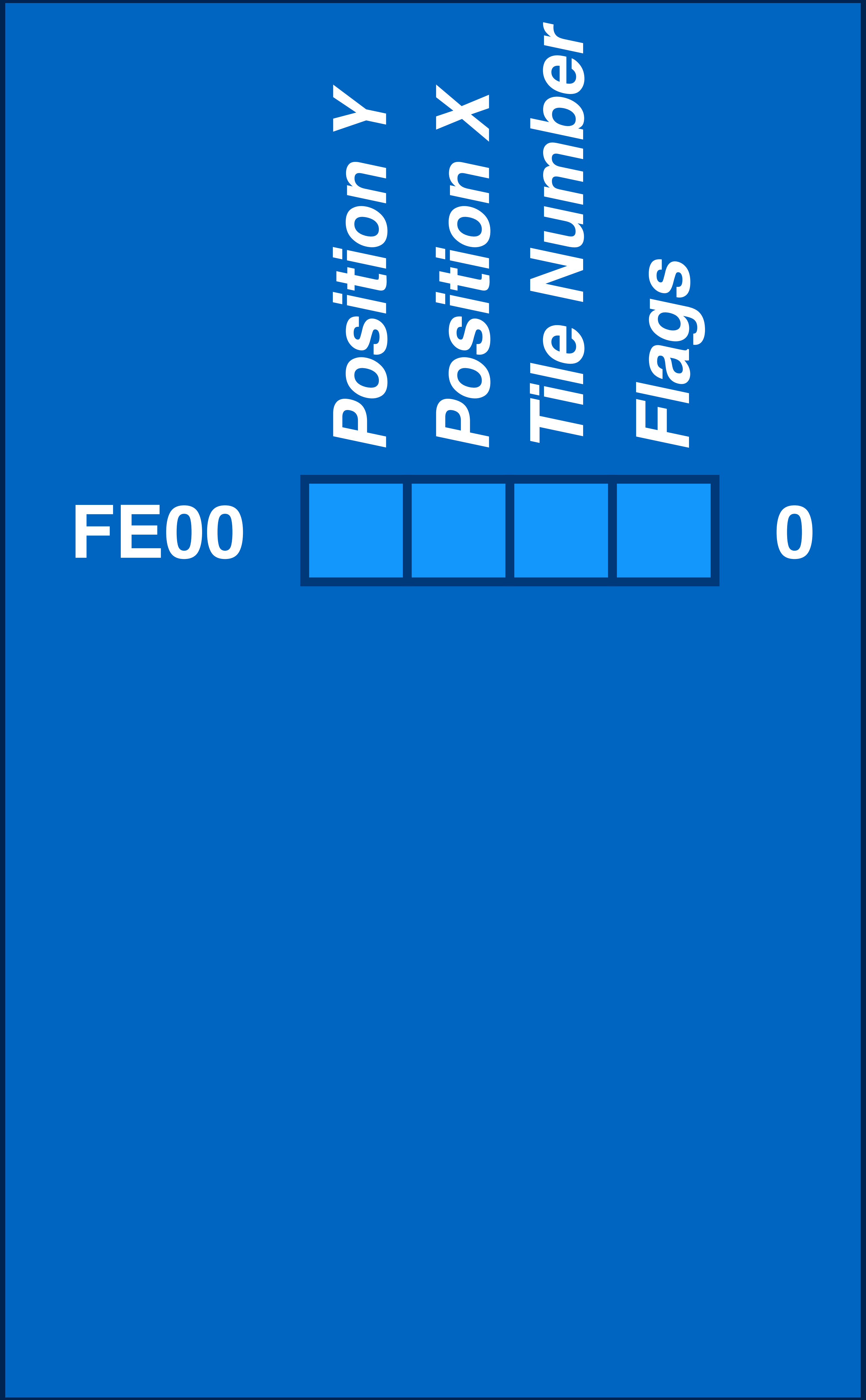
10 sprites per line

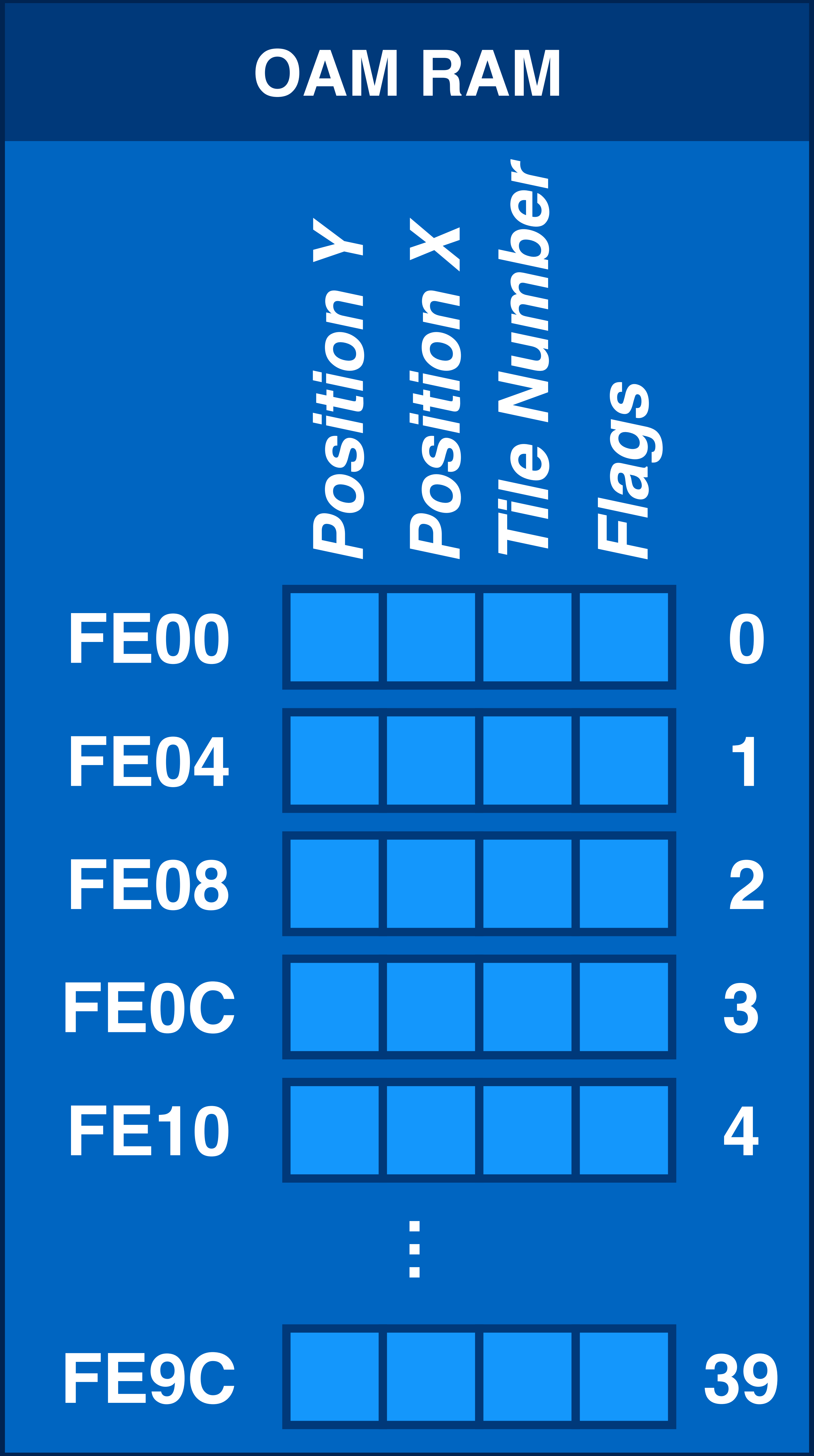
10 sprites per line

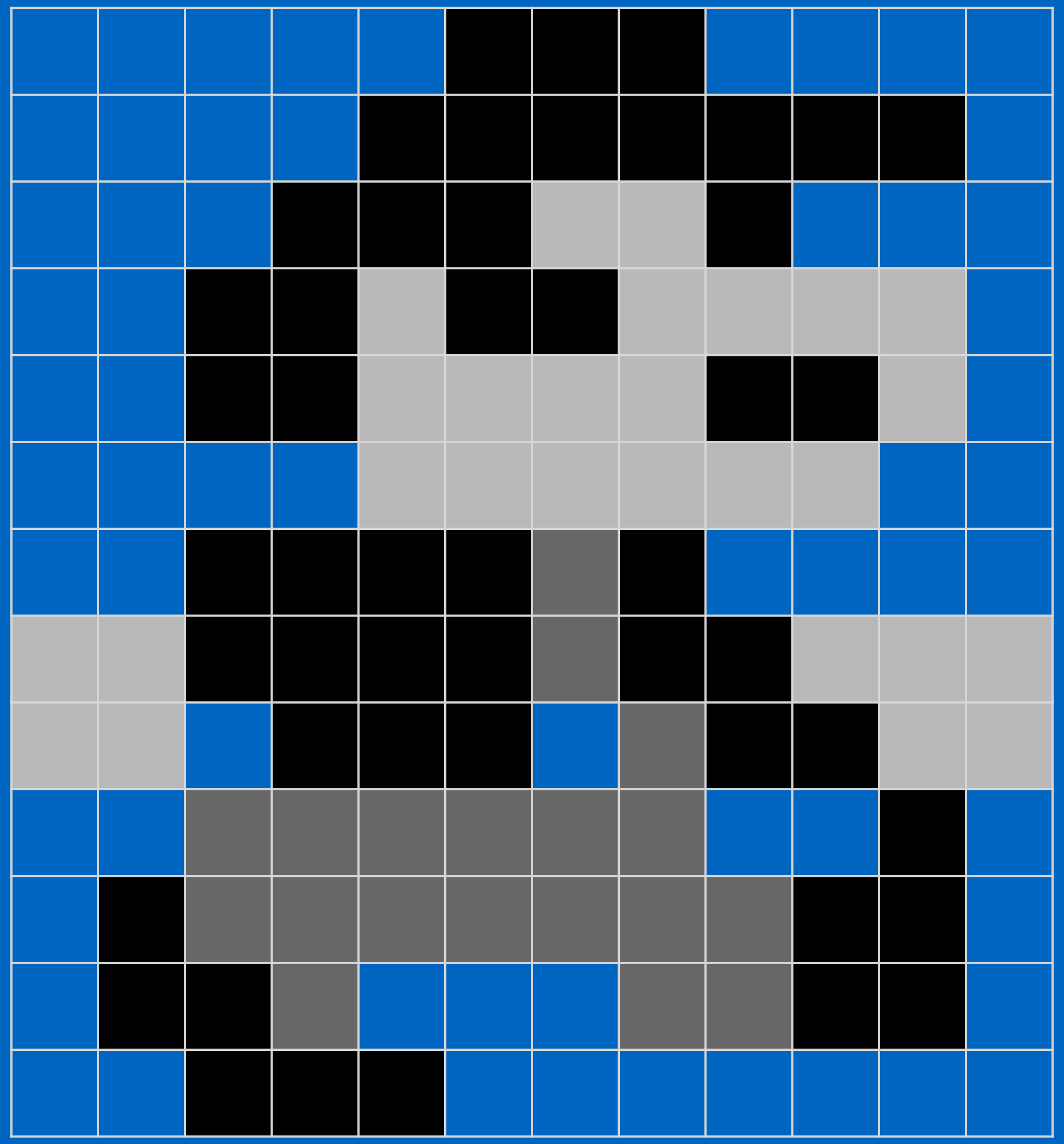


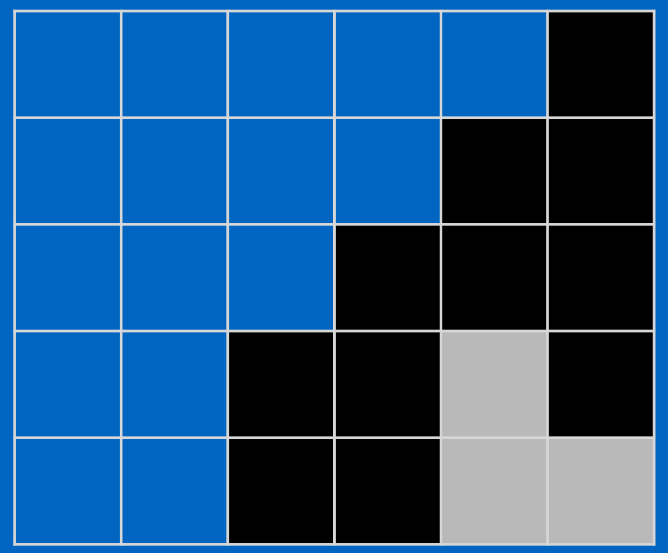
OAM Entry	
Position X	0x4D
Position Y	0x78
Tile Number	0x90
Priority	1
Flip X	☐
Flip Y	☐
Palette	0

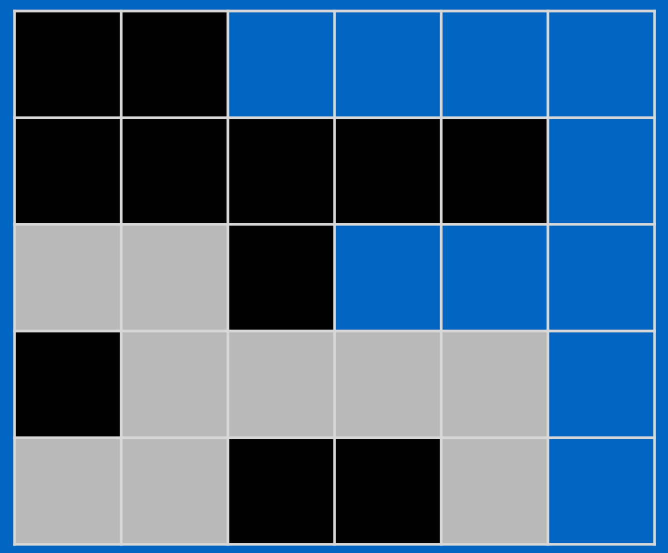

Position Y
Position X
Tile Number
Flags

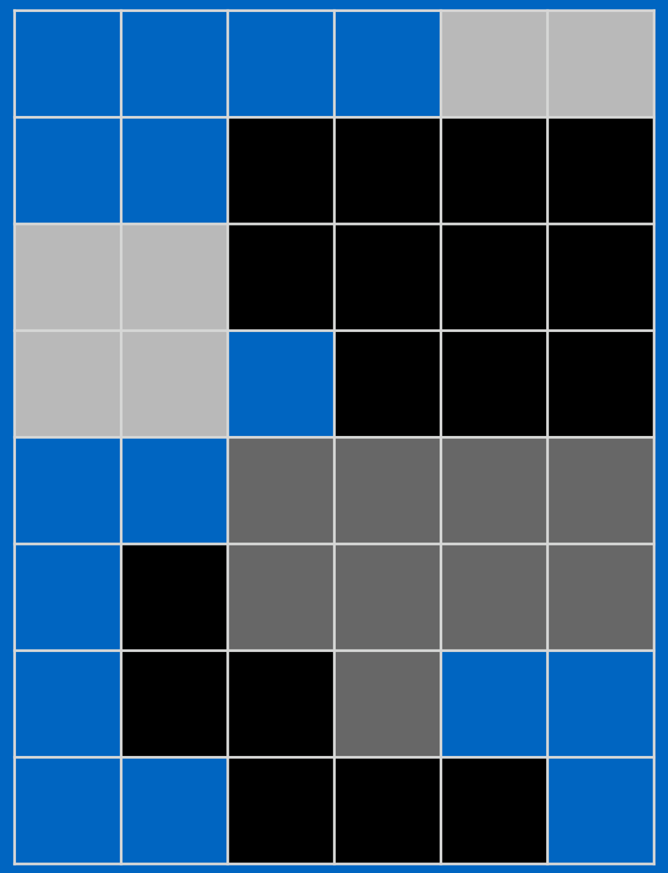

Position Y
Position X
Tile Number
Flags
FE00
0

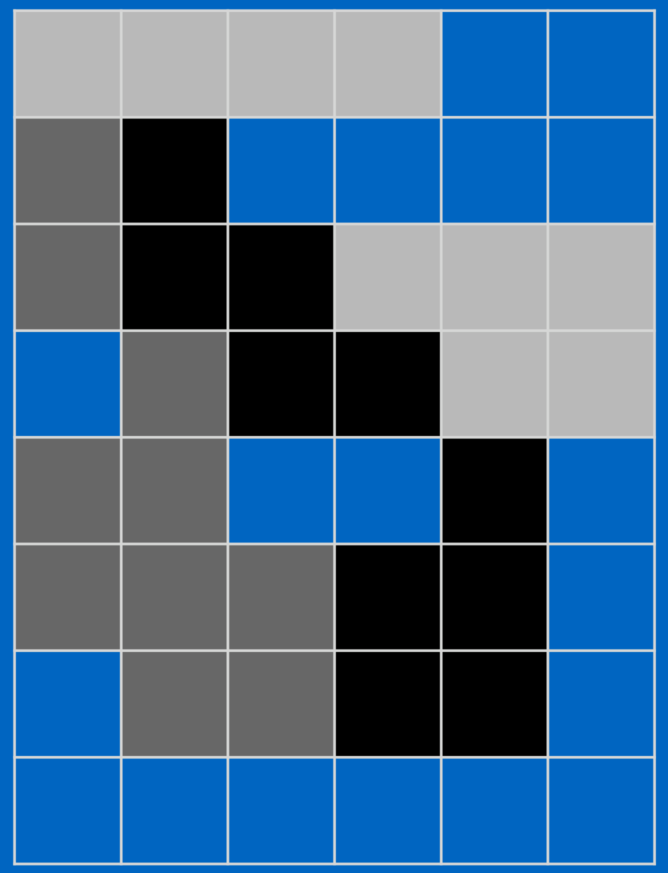

OAM RAM
Position Y
Position X
Tile Number
Flags
FE00
0
FE04
1
FE08
2
FE0C
3
FE10
4
FE9C
39



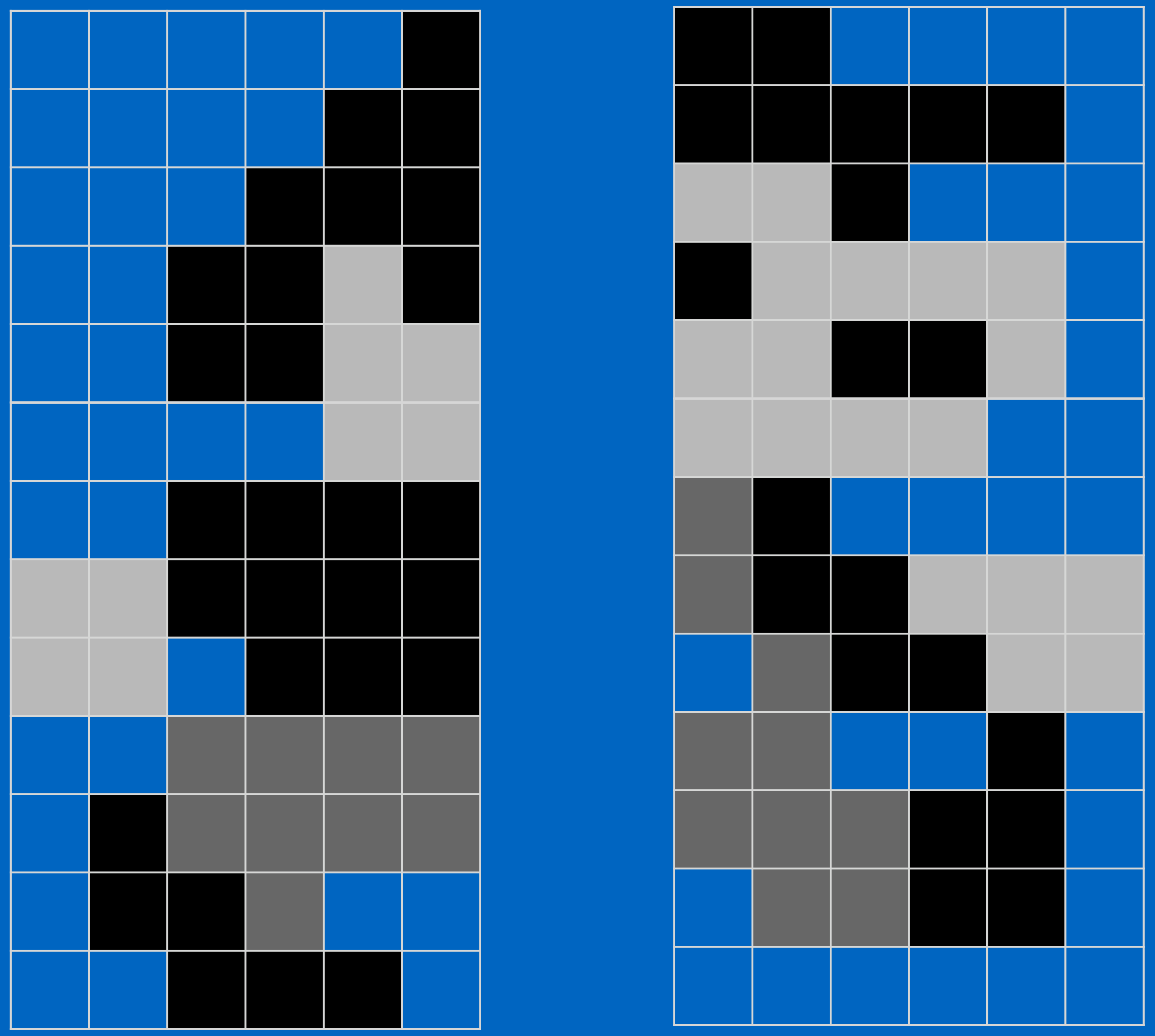








8x8 or 8x16



FF40	LCDC	LCD Control (R/W)
2		OBJ Size


OFF•ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM

Layers

FF40	LCDC	LCD Control (R/W)
7		LCD Display Enable
5		Window Enable
1		OBJ Enable
0		Background Enable

Layers

LCD Off

Off	00	01	10	11

FF40	LCDC	LCD Control (R/W)
7		LCD Display Enable
5		Window Enable
1		OBJ Enable
0		Background Enable

Layers

Off	00	01	10	11

FF40	LCDC	LCD Control (R/W)
7		LCD Display Enable
5		Window Enable
1		OBJ Enable
0		Background Enable

Layers

Off	00	01	10	11

FF40	LCDC	LCD Control (R/W)
7		LCD Display Enable
5		Window Enable
1		OBJ Enable
0		Background Enable

Layers

Off	00	01	10	11

FF40	LCDC	LCD Control (R/W)
7		LCD Display Enable
5		Window Enable
1		OBJ Enable
0		Background Enable

Layers

Off	00	01	10	11

FF40	LCDC	LCD Control (R/W)
7		LCD Display Enable
5		Window Enable
1		OBJ Enable
0		Background Enable

VRAM Memory Map

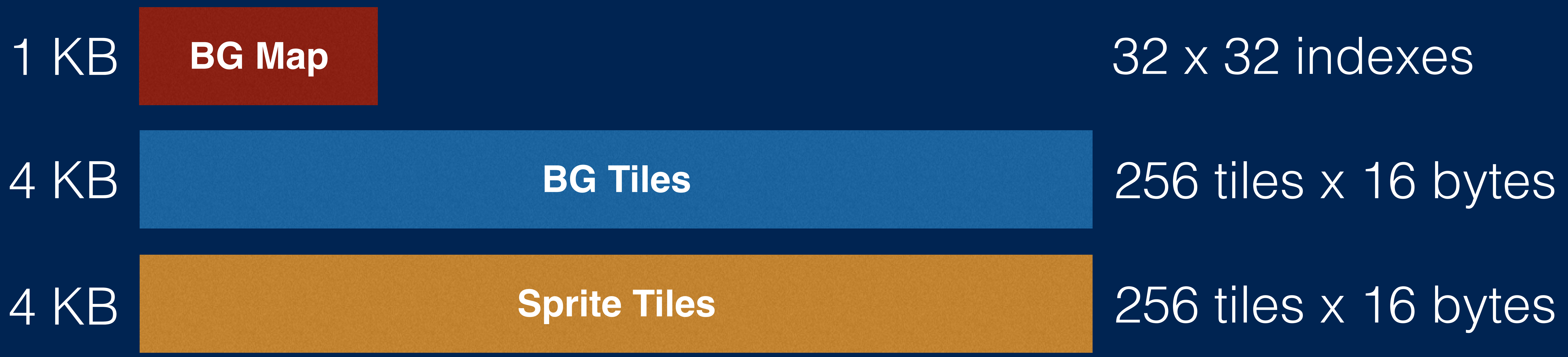
VRAM Memory Map

4 KB | Sprite Tiles | 256 tiles x 16 bytes

VRAM Memory Map

4 KB	**BG Tiles**	256 tiles x 16 bytes
4 KB	**Sprite Tiles**	256 tiles x 16 bytes

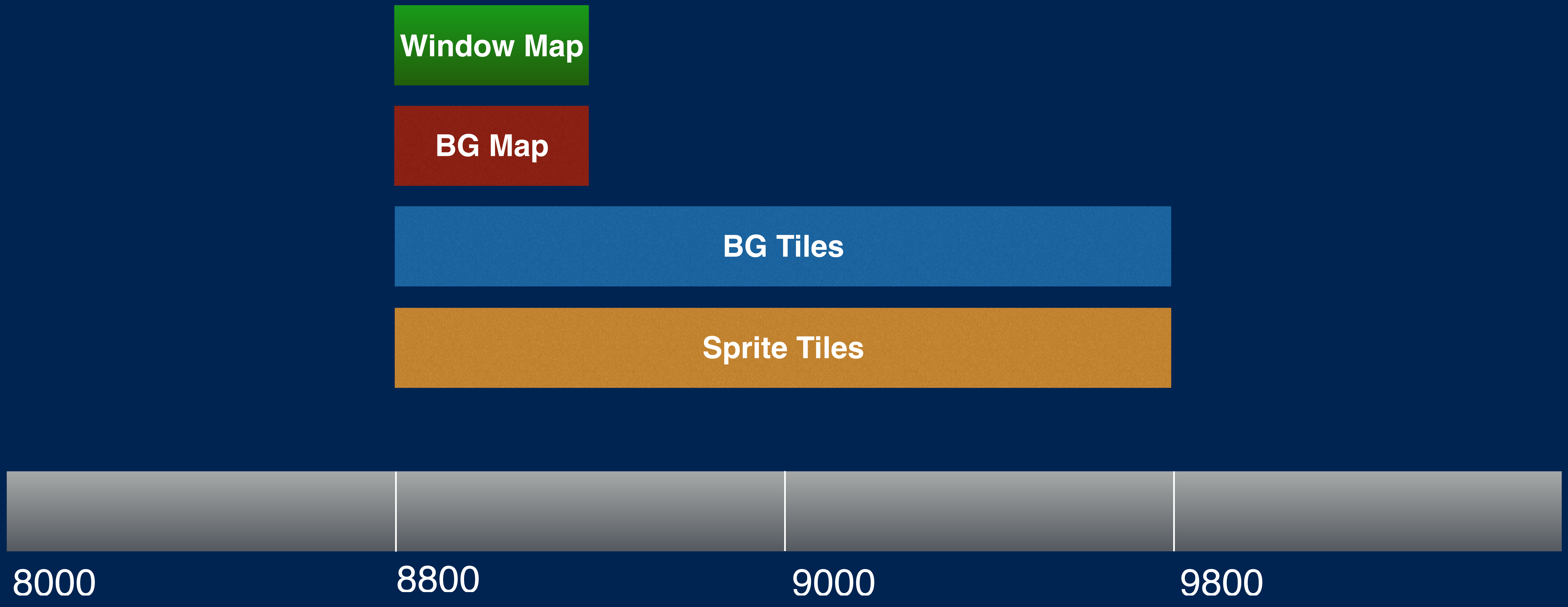
VRAM Memory Map



VRAM Memory Map

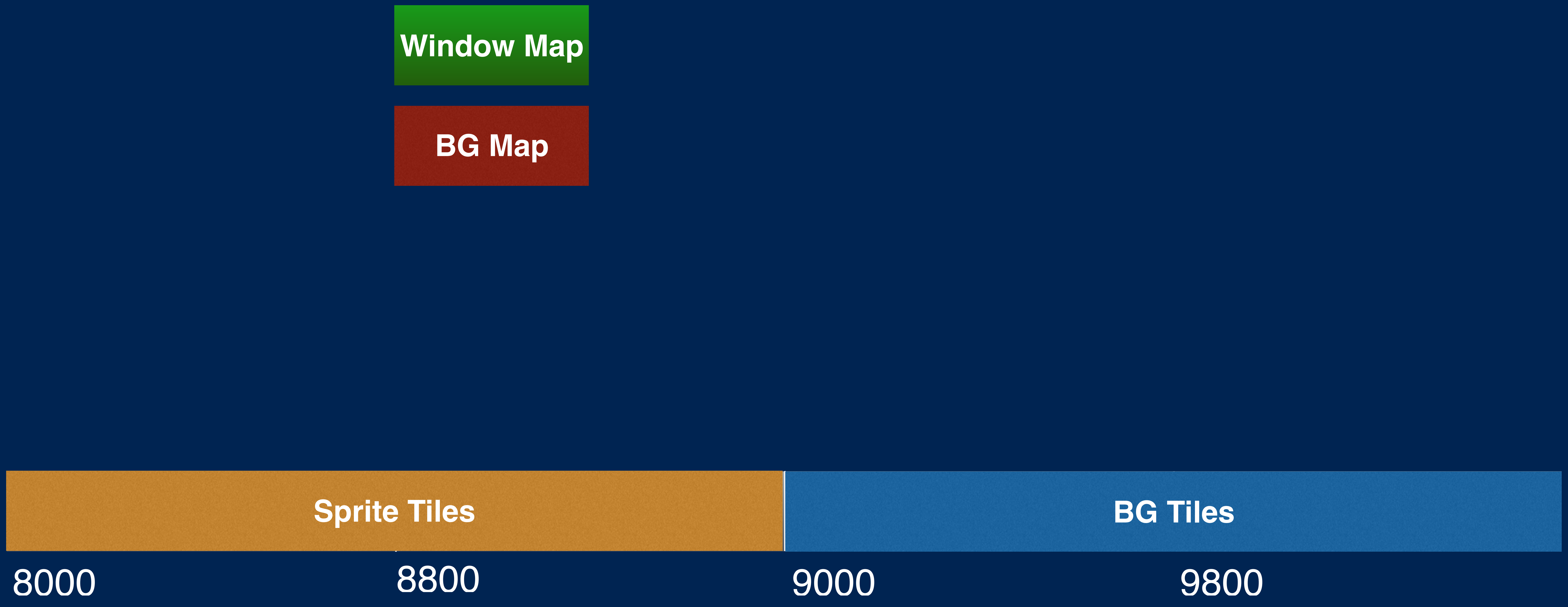
1 KB	Window Map	32 x 32 indexes
1 KB	BG Map	32 x 32 indexes
4 KB	BG Tiles	256 tiles x 16 bytes
4 KB	Sprite Tiles	256 tiles x 16 bytes

VRAM Memory Map

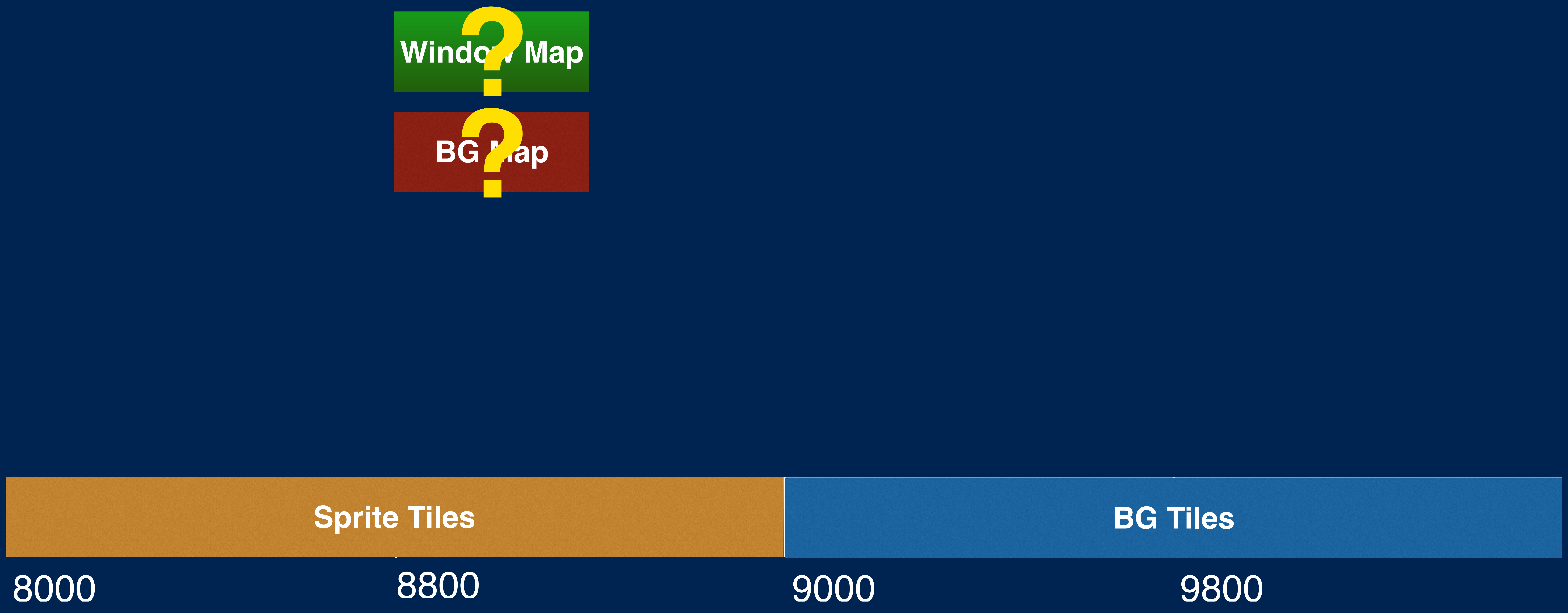



VRAM Memory Map
Window Map
BG Map
BG Tiles
Sprite Tiles
8000
8800
9000
9800

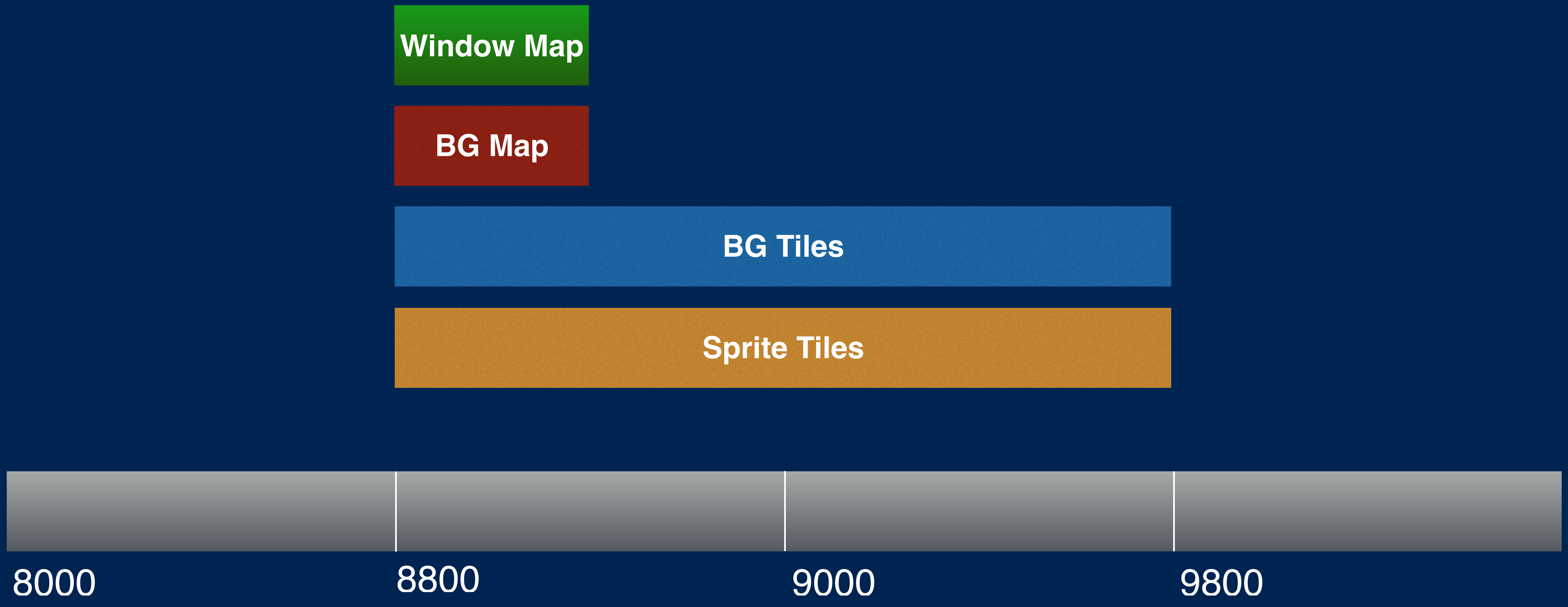
VRAM Memory Map



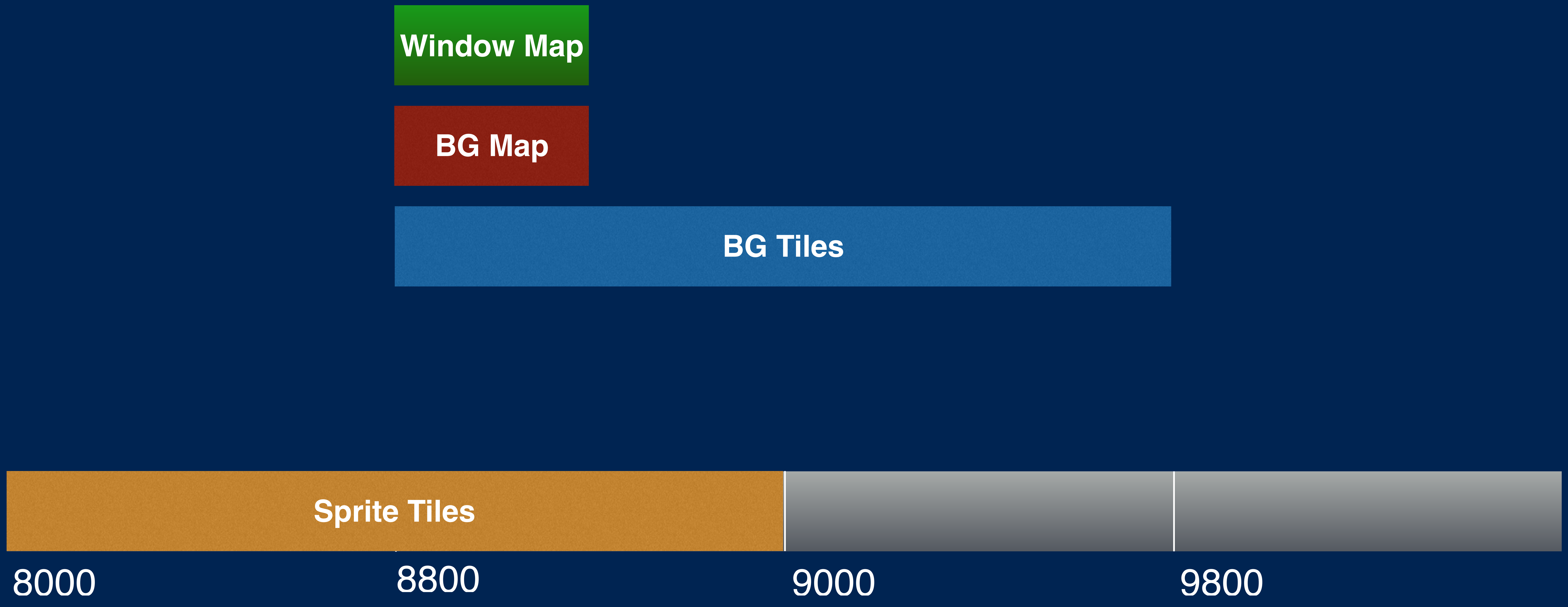
VRAM Memory Map



VRAM Memory Map



VRAM Memory Map



VRAM Memory Map

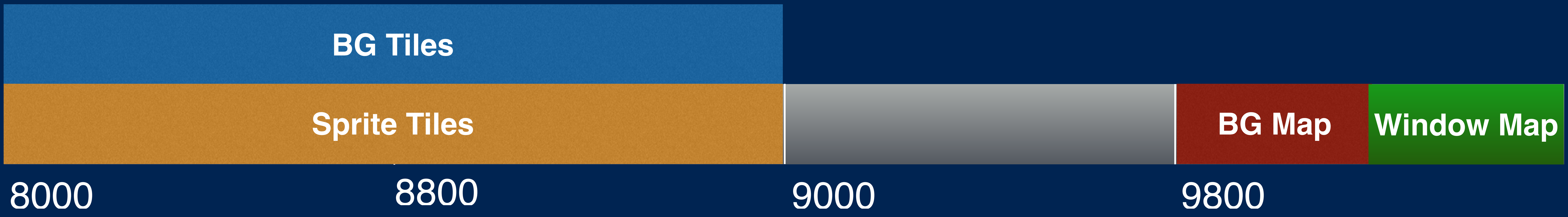
VRAM Memory Map

VRAM Memory Map

FF40	LCDC	LCD Control (R/W)
6		Window Tile Map Address
4		BG & Window Tile Data Address
3		BG Tile Map Address

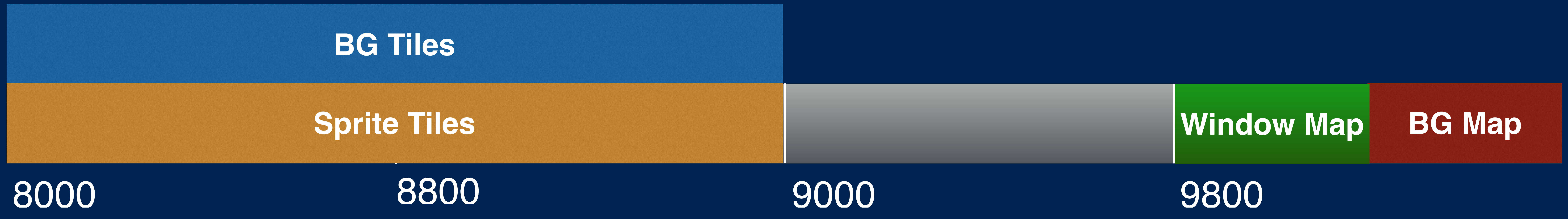
VRAM Memory Map

FF40	LCDC	LCD Control (R/W)
6		Window Tile Map Address
4		BG & Window Tile Data Address
3		BG Tile Map Address



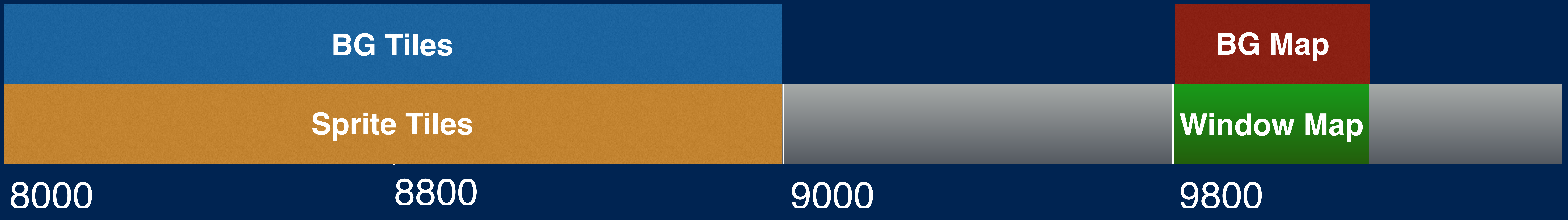
VRAM Memory Map

FF40	LCDC	LCD Control (R/W)
6		Window Tile Map Address
4		BG & Window Tile Data Address
3		BG Tile Map Address



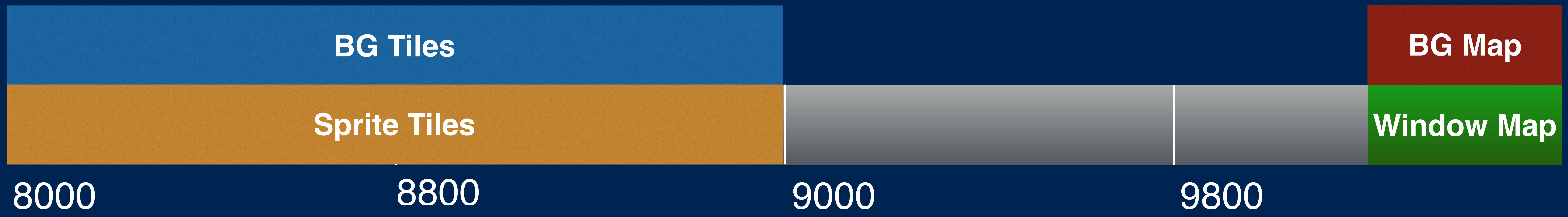
VRAM Memory Map

FF40	LCDC	LCD Control (R/W)
6		Window Tile Map Address
4		BG & Window Tile Data Address
3		BG Tile Map Address



VRAM Memory Map

FF40	LCDC	LCD Control (R/W)
6		Window Tile Map Address
4		BG & Window Tile Data Address
3		BG Tile Map Address




8000
8800
9000
9800
Sprite Tiles
BG Tiles

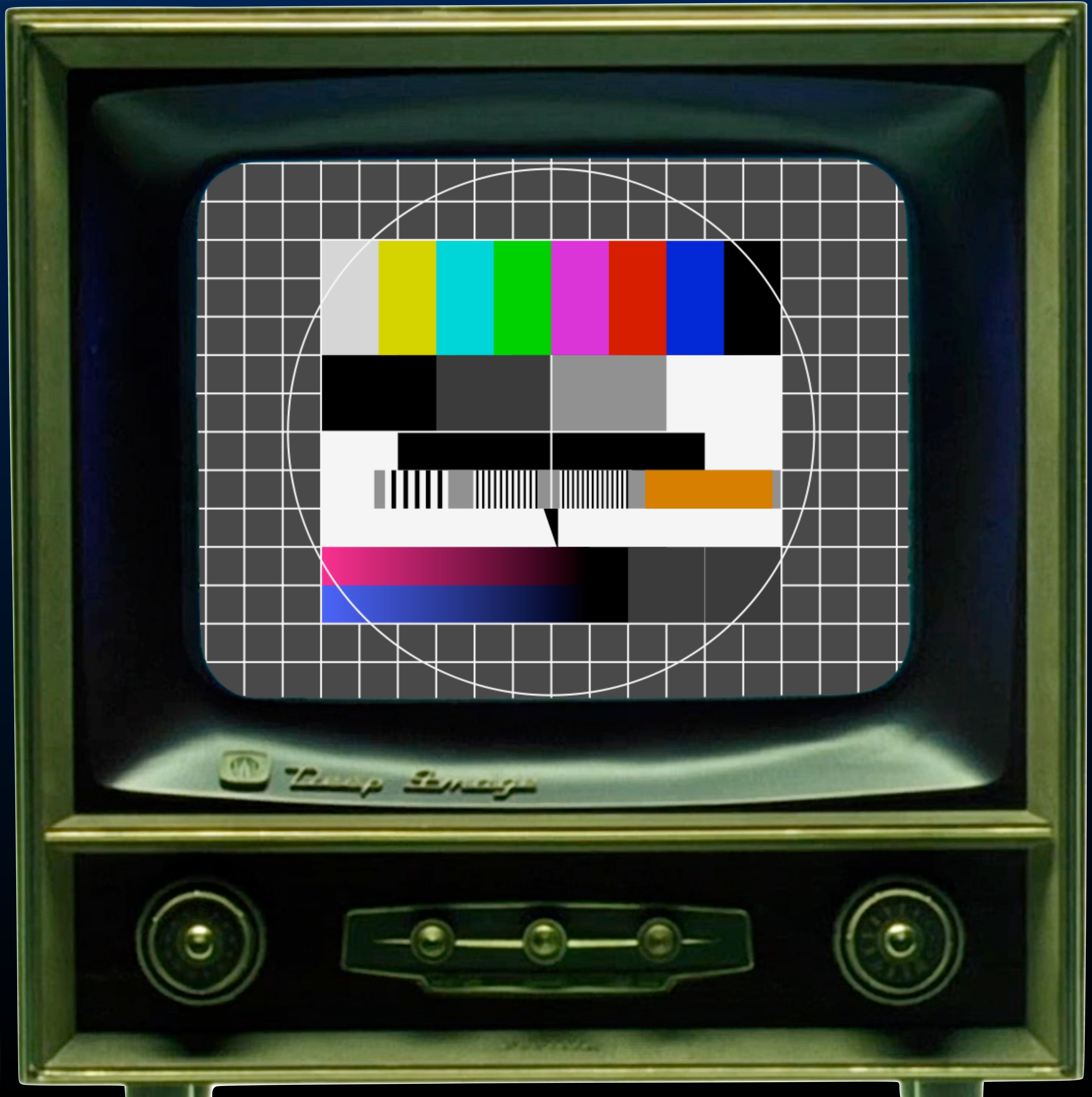

8000
8800
9000
9800
Sprite Tiles
BG Tiles

Pixel Processing Unit

Pixel Processing Unit

Vertical Timing

Timing




OFF·ON
DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM

Timing


OFF · ON
DOT MATRIX WITH STEREO SOUND
MARIOx02 WORLD TIME
0 x00 1-1 330
BATTERY
Nintendo GAME BOY TM


DOT MATRIX WITH STEREO SOUND
BATTERY
TIME 0'09"04 LAP 01
F-1GP
TROUBLE PARTS
121

BATTERY
TIME 0'09"04 LAP 01
F-1GP
N
1
2
3
4
TROUBLE
PARTS
121


DOT MATRIX WITH STEREO SOUND
TIME 0'09"04 LAP 01
F-1GP
BATTERY
TROUBLE
PARTS
N
1
2
3
4
121 km/h


DOT MATRIX WITH STEREO SOUND
BATTERY
TIME 0'09"04 LAP 01
F-1GP
N
1
2
3
4
TROUBLE
PARTS
121 km/h


DOT MATRIX WITH STEREO SOUND
TIME 0'09"04 LAP 01
F-1GP
BATTERY
N
1
2
3
4
TROUBLE
PARTS
121 km/h


DOT MATRIX WITH STEREO SOUND
BATTERY
TIME 0'09"04 LAP 01
F-1GP
F-1GP
TROUBLE PARTS
N
1
2
3
4
121 km/h


DOT MATRIX WITH STEREO SOUND
BATTERY

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

SCX = 0

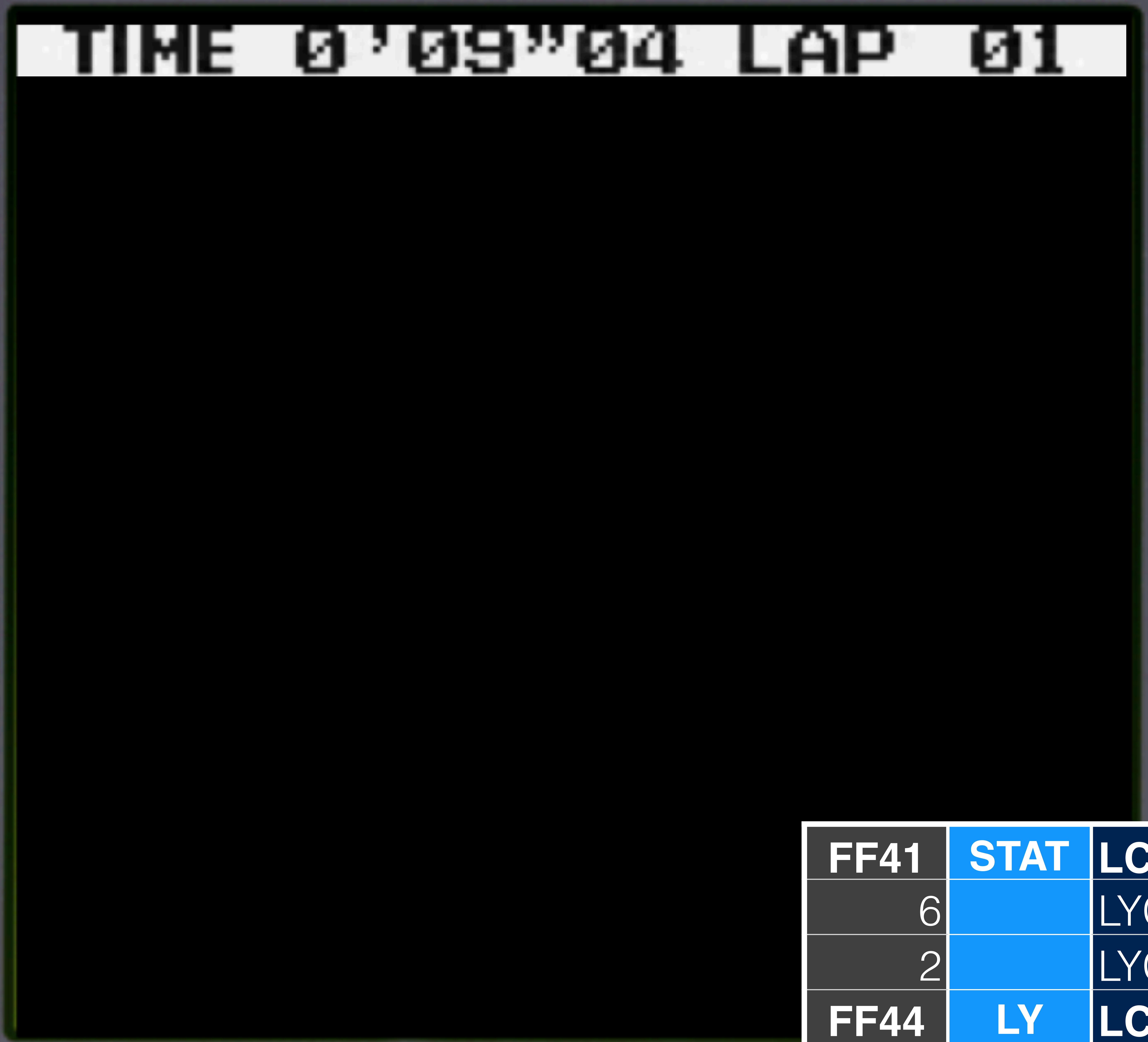
FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

LYC Interrupt

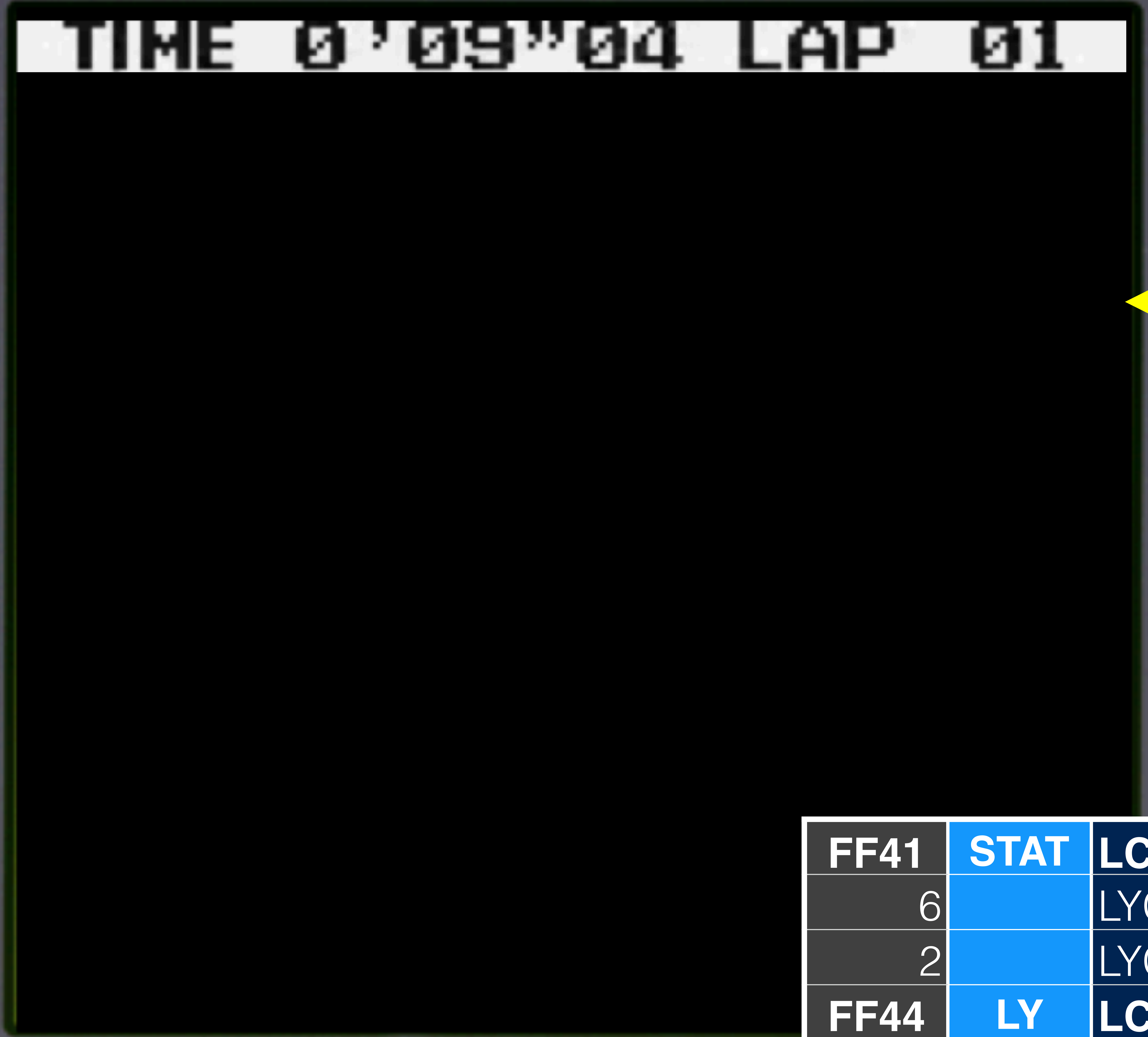
SCX = 23



FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

LYC Interrupt

SCX = 23



LYC = 42

BATTERY

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

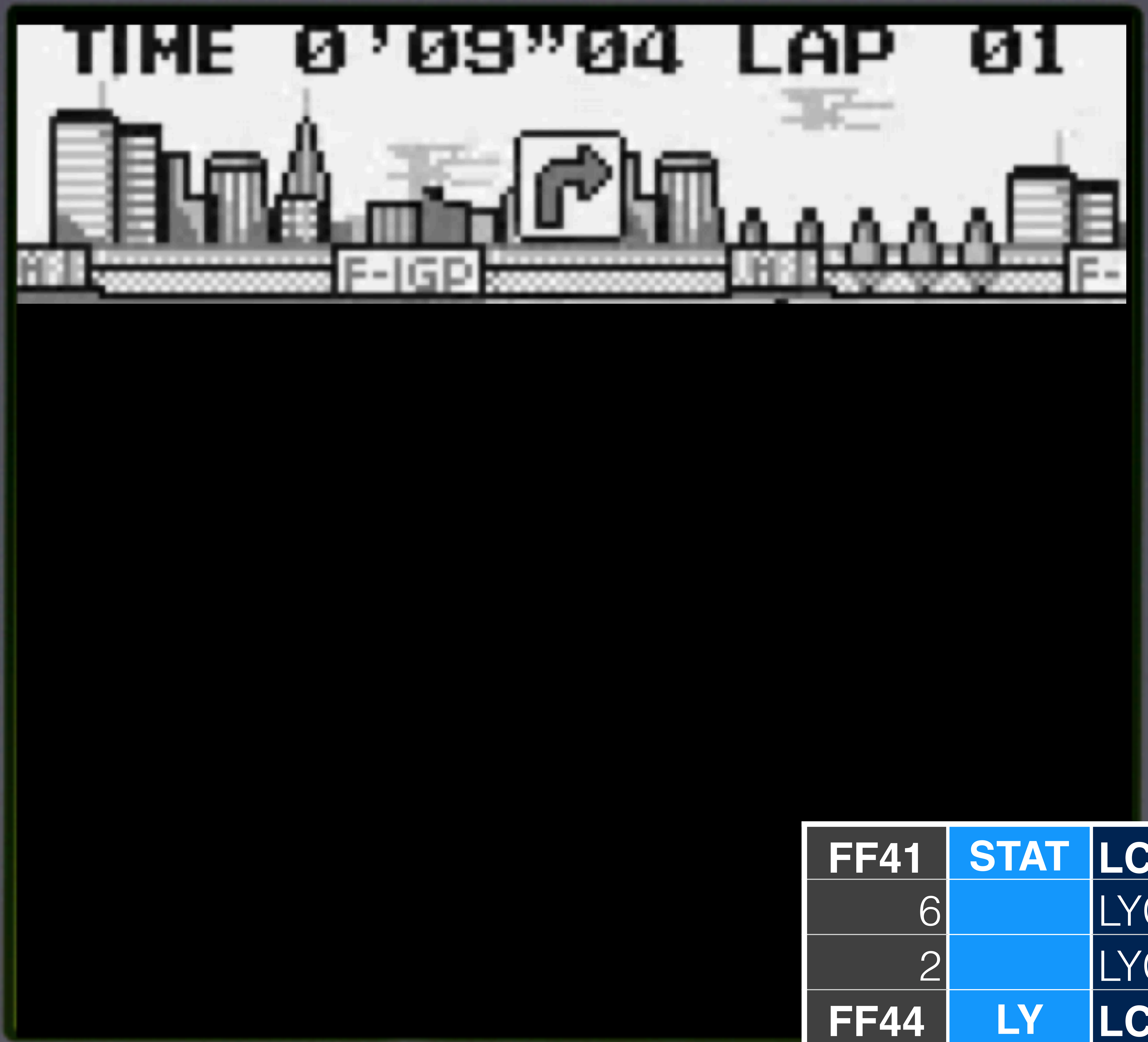
LYC Interrupt

SCX = 23

LYC = 42

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

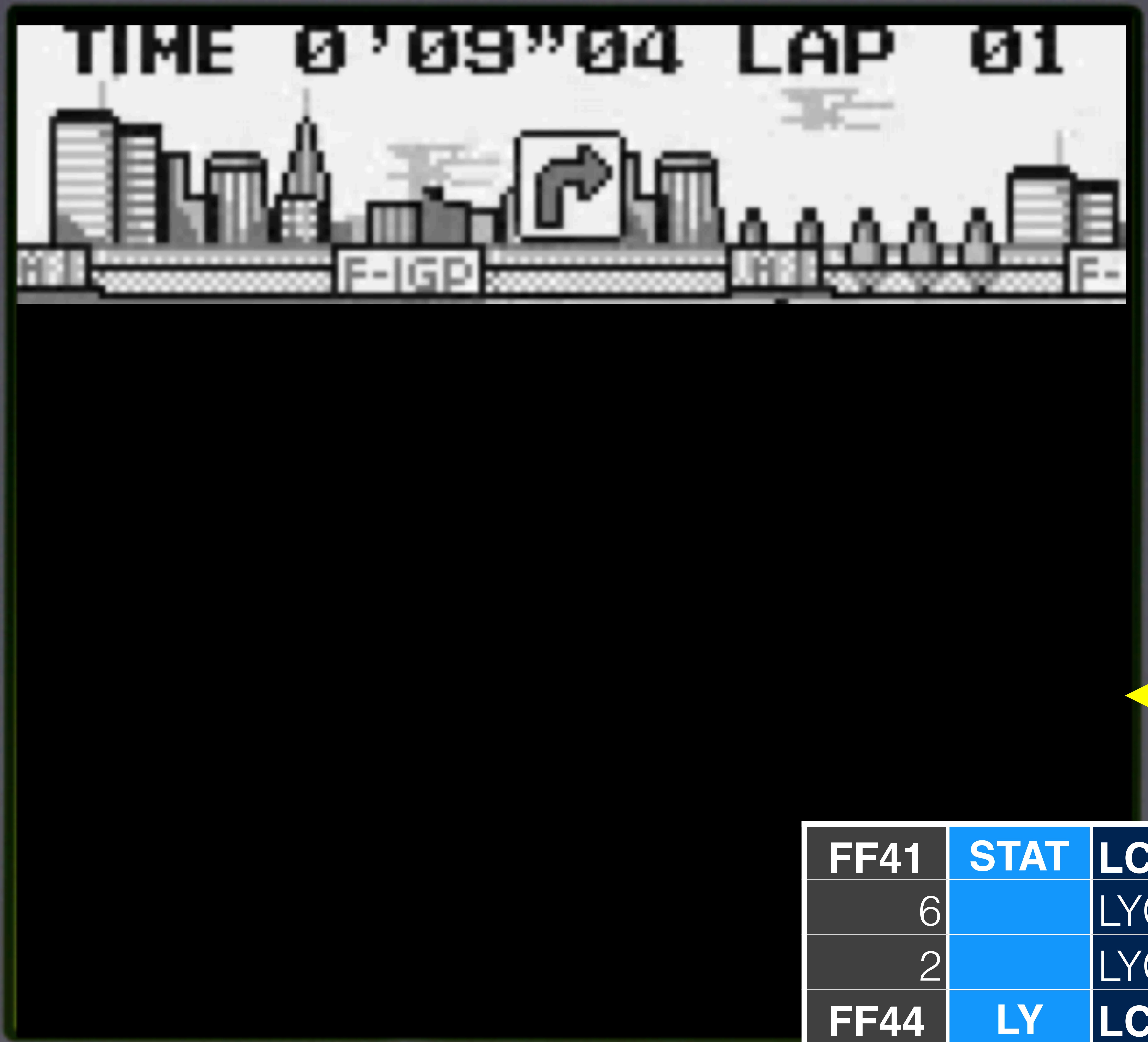
LYC Interrupt



SCX = ?

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

LYC Interrupt



SCX = ?

LYC = 88

FF41	STAT	LCDC Status (R/W)
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	LY	LCDC Y-Coordinate (R)
FF45	LYC	LY Compare (R/W)

LYC Interrupt

SCX = ?

LYC = 88

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

LYC Interrupt

SCX = 0

FF41	**STAT**	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	**LY**	**LCDC Y-Coordinate (R)**
FF45	**LYC**	**LY Compare (R/W)**

LYC Interrupt

LYC = 8

SCX = 0

FF41	STAT	LCDC Status (R/W)
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	LY	LCDC Y-Coordinate (R)
FF45	LYC	LY Compare (R/W)

LYC Interrupt

LYC = 8

SCX = 0

FF41	STAT	**LCDC Status (R/W)**
6		LYC=LY Interrupt
2		LYC=LY Flag
FF44	LY	**LCDC Y-Coordinate (R)**
FF45	LYC	**LY Compare (R/W)**


DOT MATRIX WITH STEREO SOUND
BATTERY
FAULT


BATTERY


LCY = 0
BATTERY
FAULT


LCY = 0
LCDC.WINON = 1
BATTERY


LCY = 0
LCDC.WINON = 1
LCY = 40
BATTERY


FAULT
BATTERY
LCY = 0
LCDC.WINON = 1
LCY = 40
LCDC.WINON = 0


MARIO×02 WORLD TIME
0 ×00 1-1 400


TOTAL
P. 0
HP
TIME 000


TIME 0'09"04 LAP 01
F-1GP
N
1
2
3
4
TROUBLE PARTS
121


SC 000000 HI 000000


MARIO×02 WORLD TIME
0 ×00 1-1 400


TOTAL
P. 0
HP
TIME 000

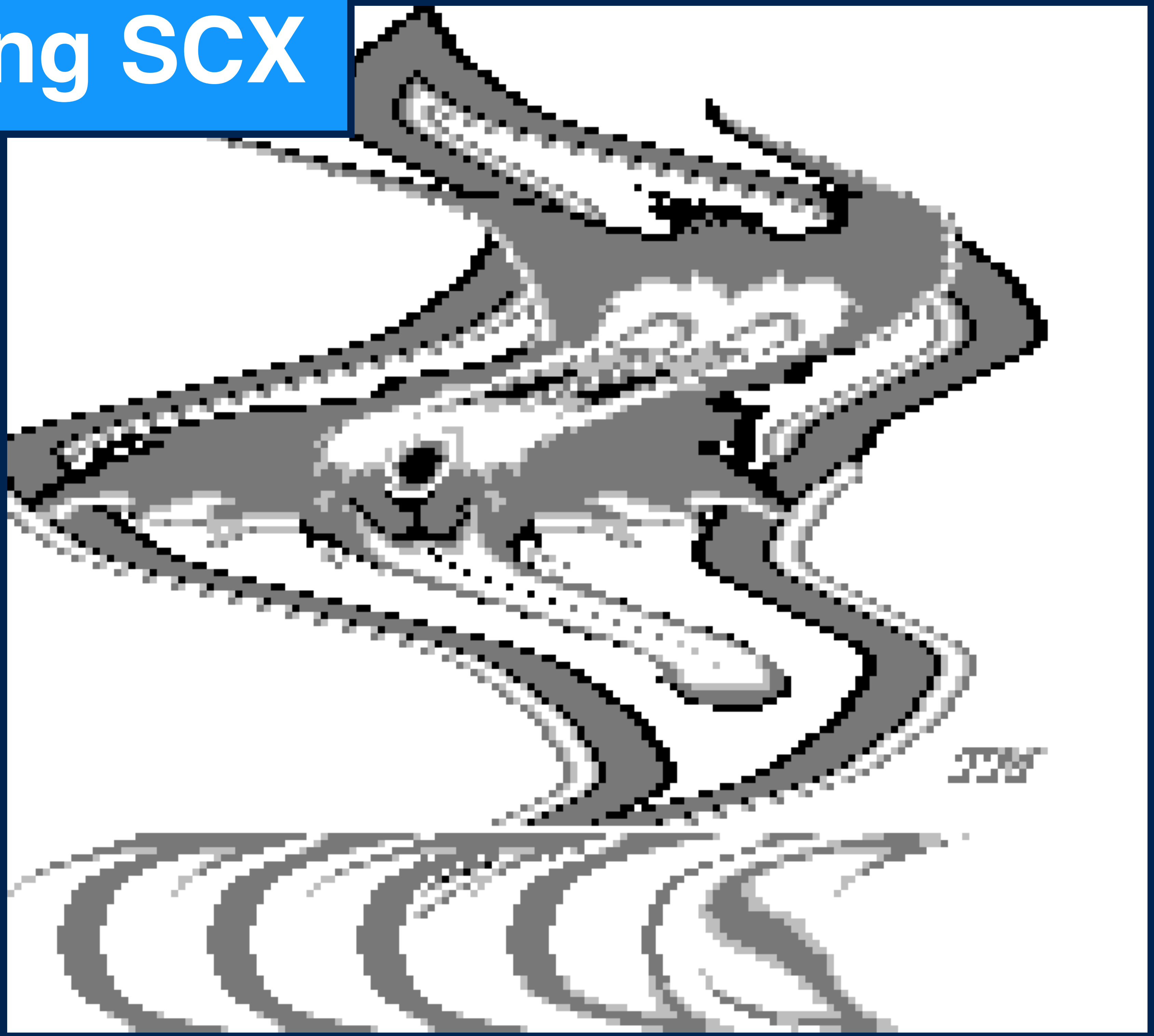

TIME 0'09"04 LAP 01
F-1GP
TROUBLE PARTS
121 km/h

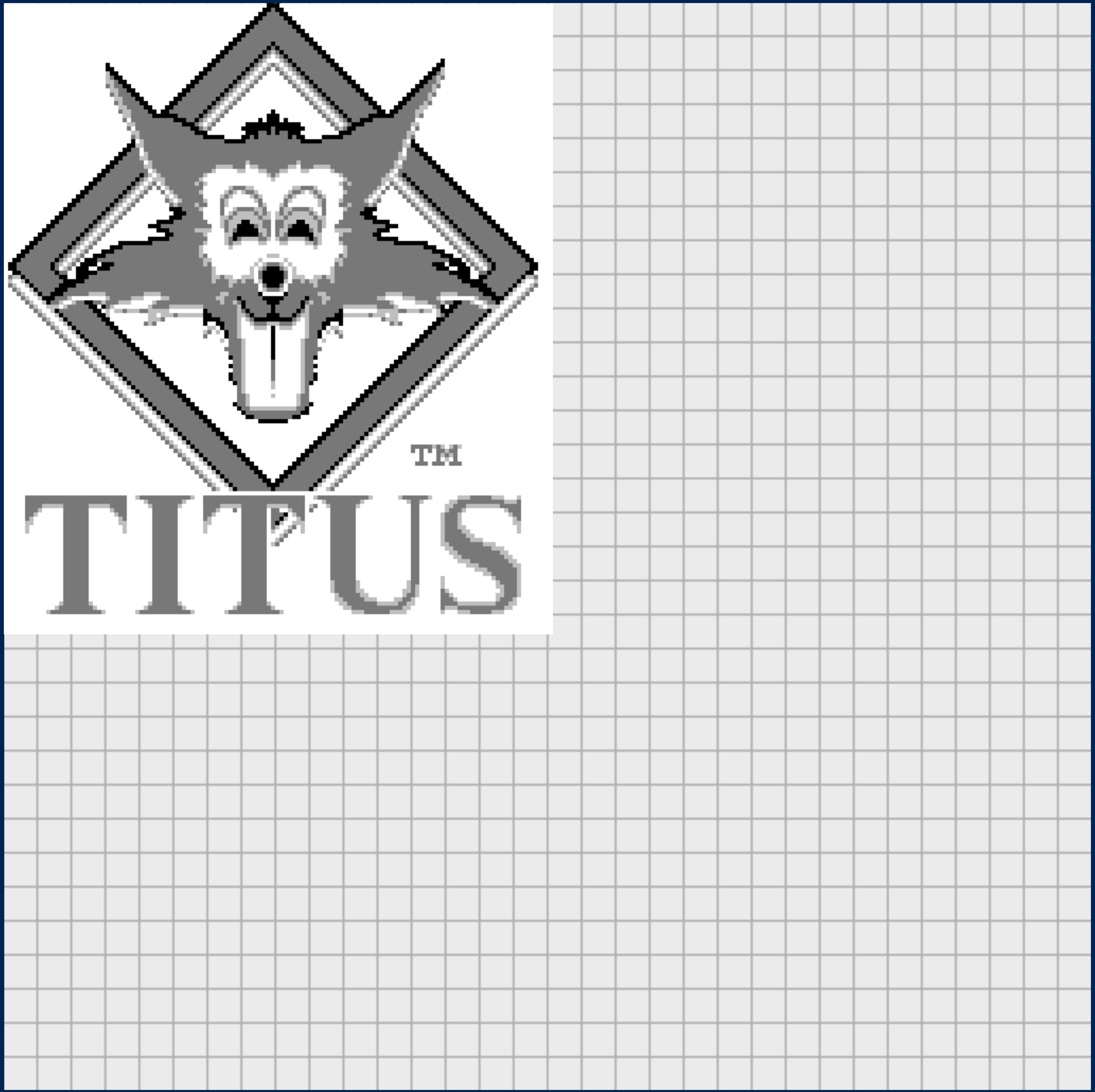

SC 000000 HI 000000

TITUS™

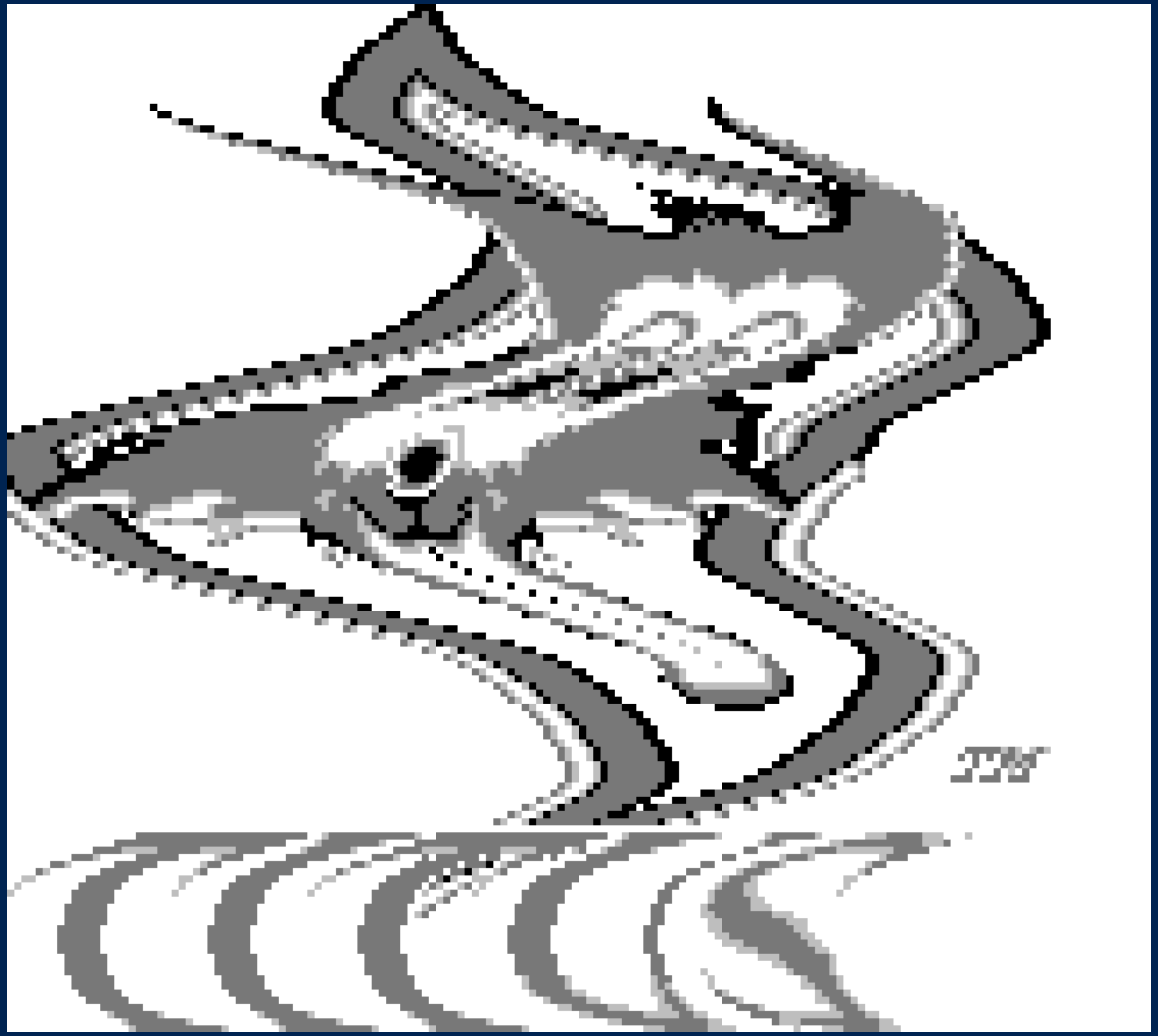
TITUS

Changing SCX

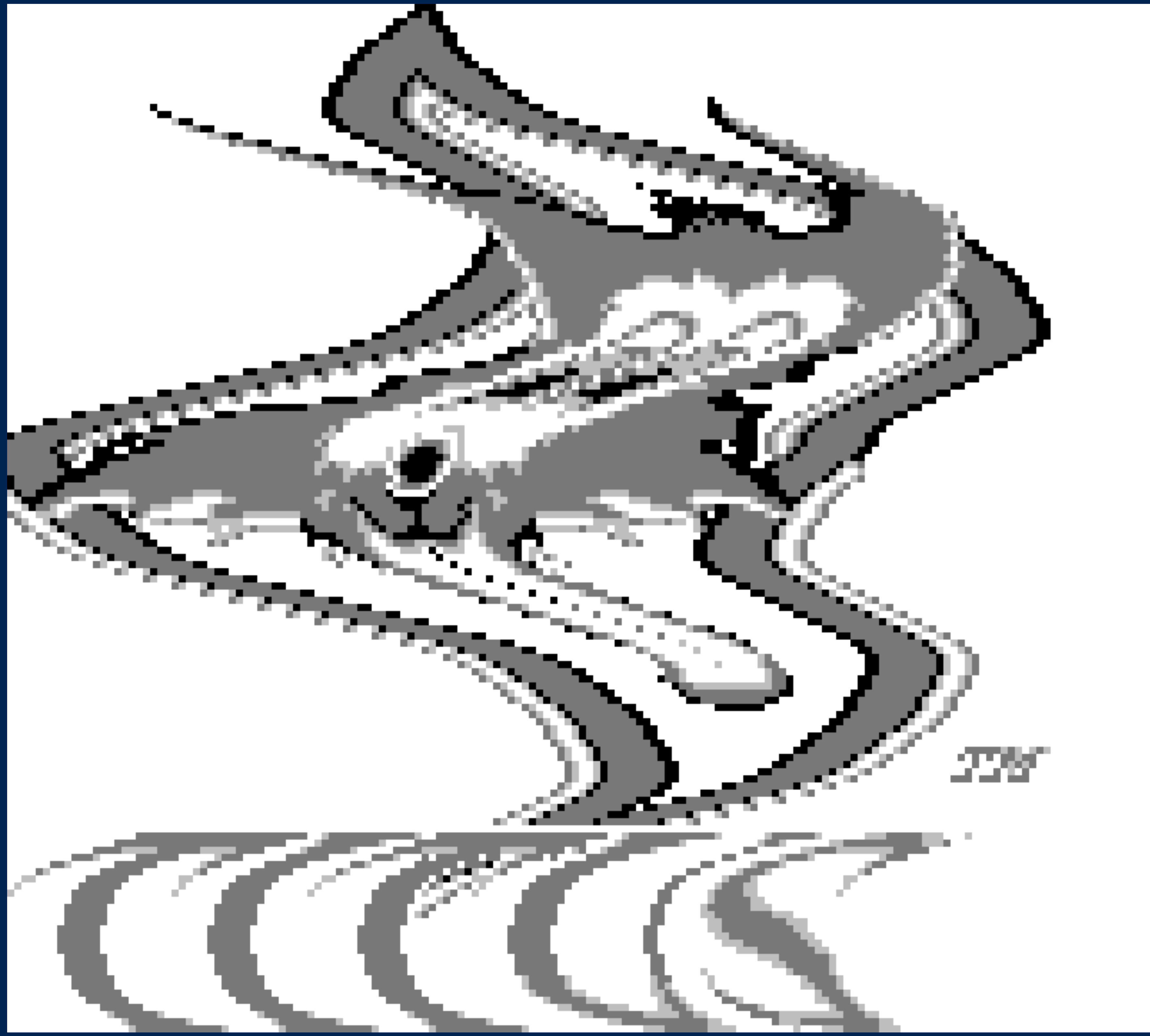




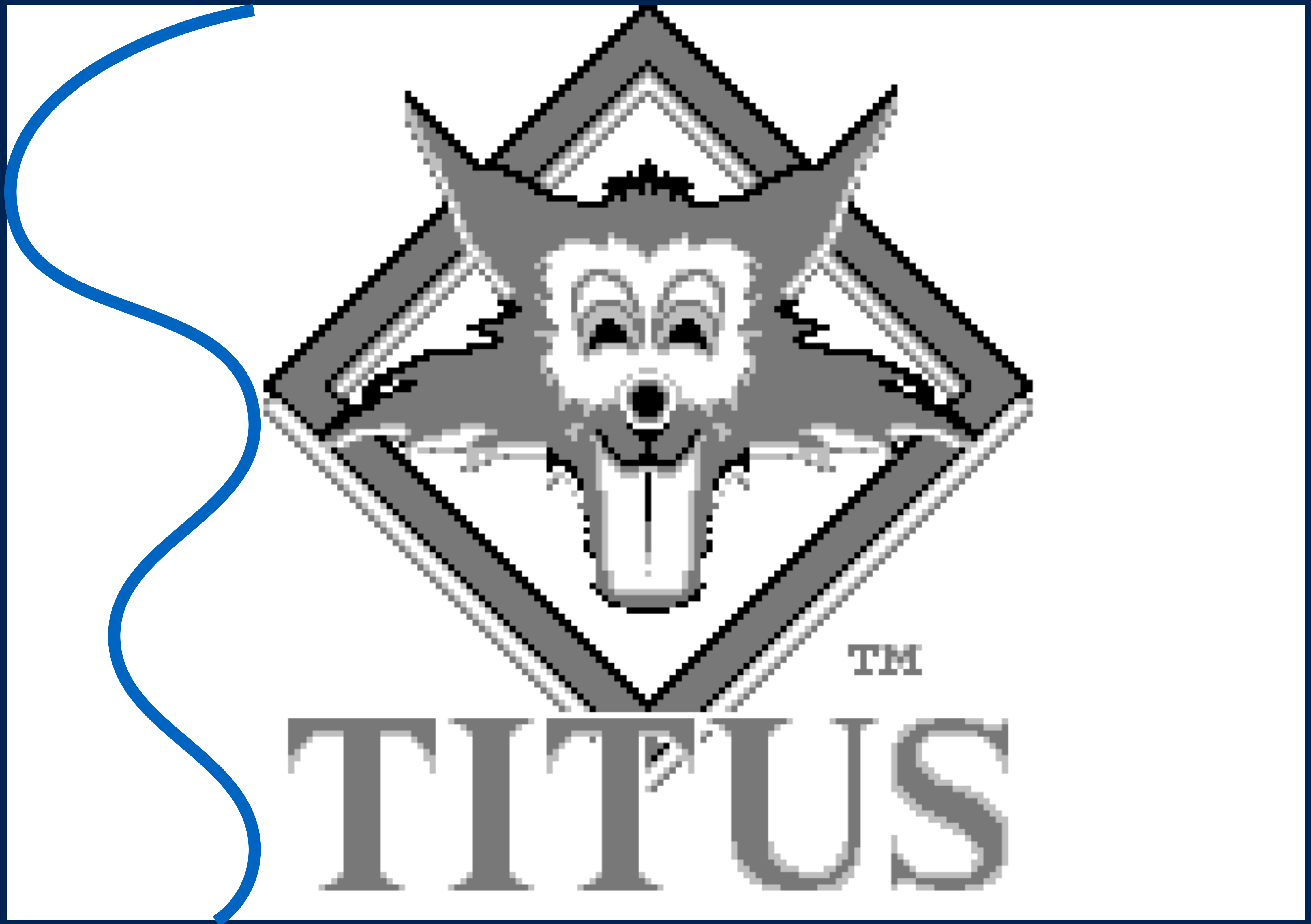
Changing SCX

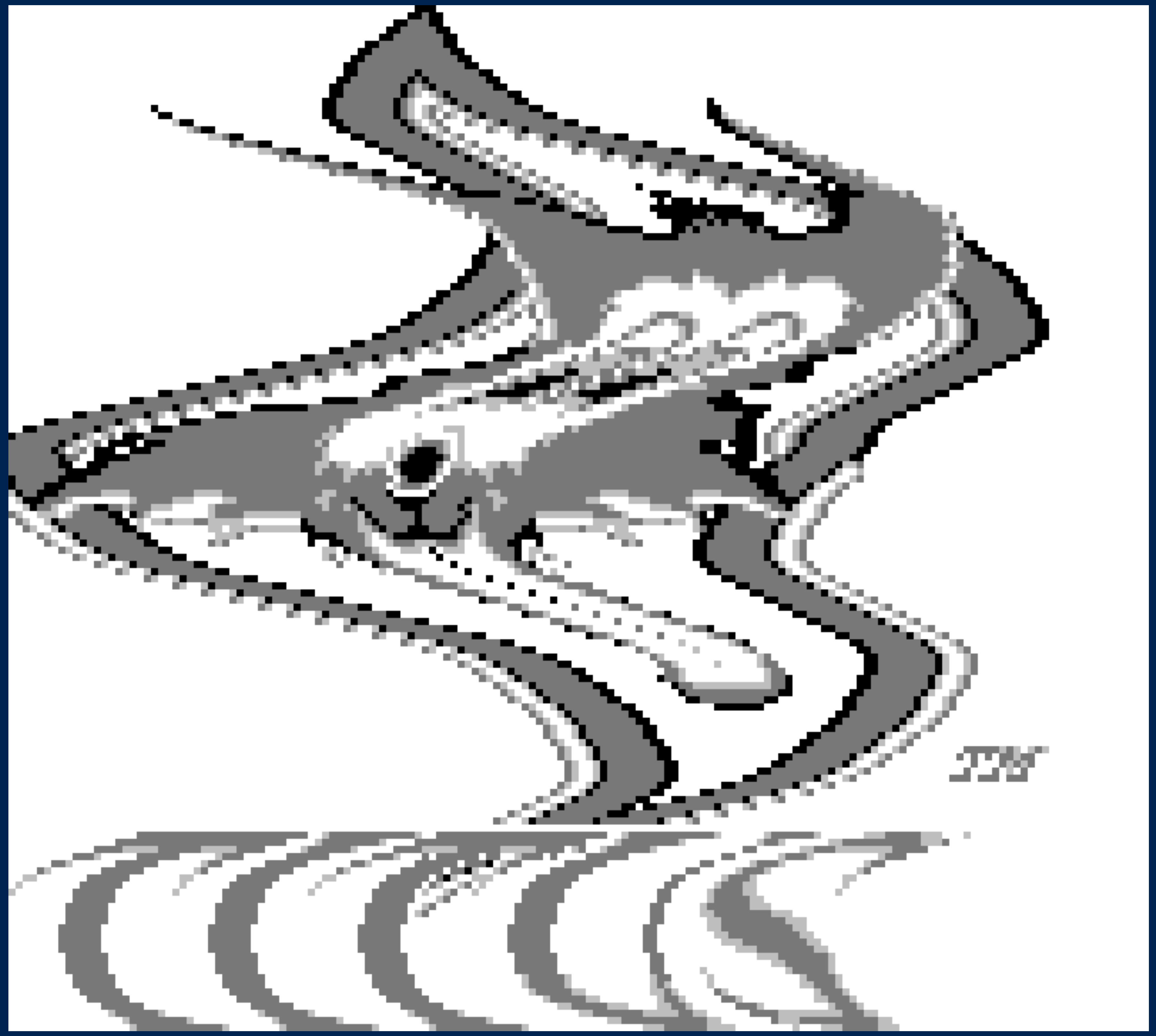


Changing SCX

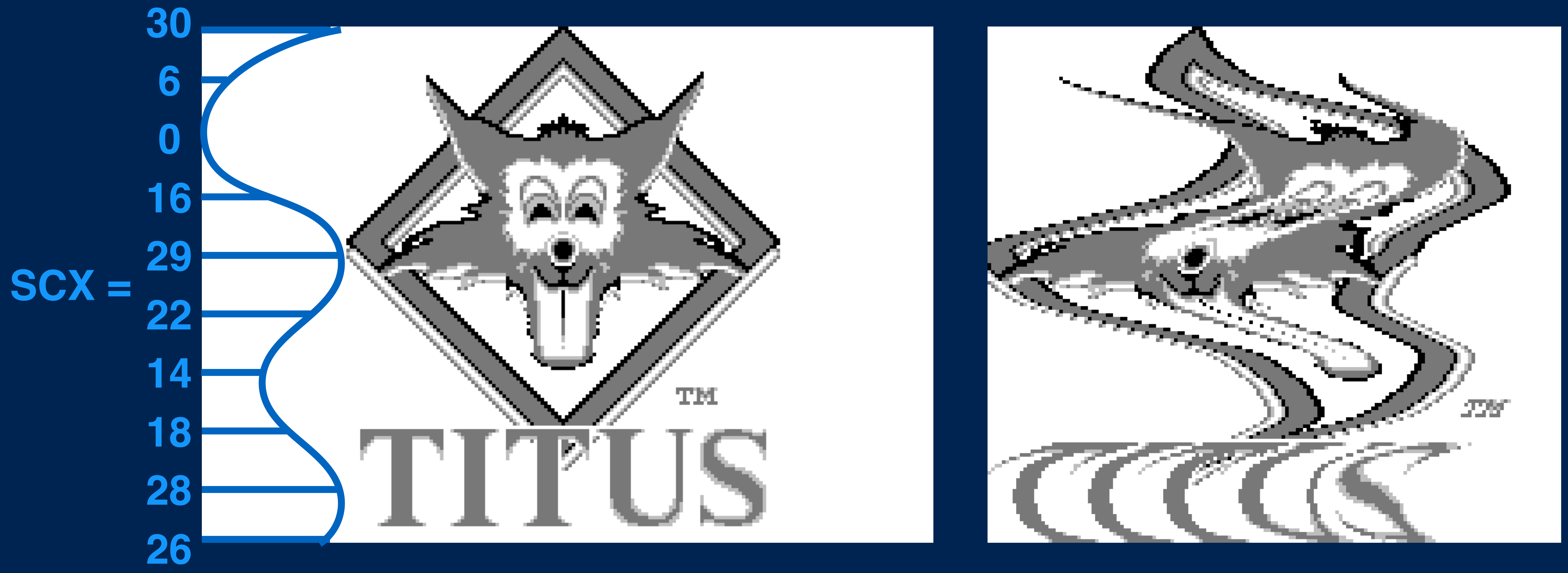


Changing SCX





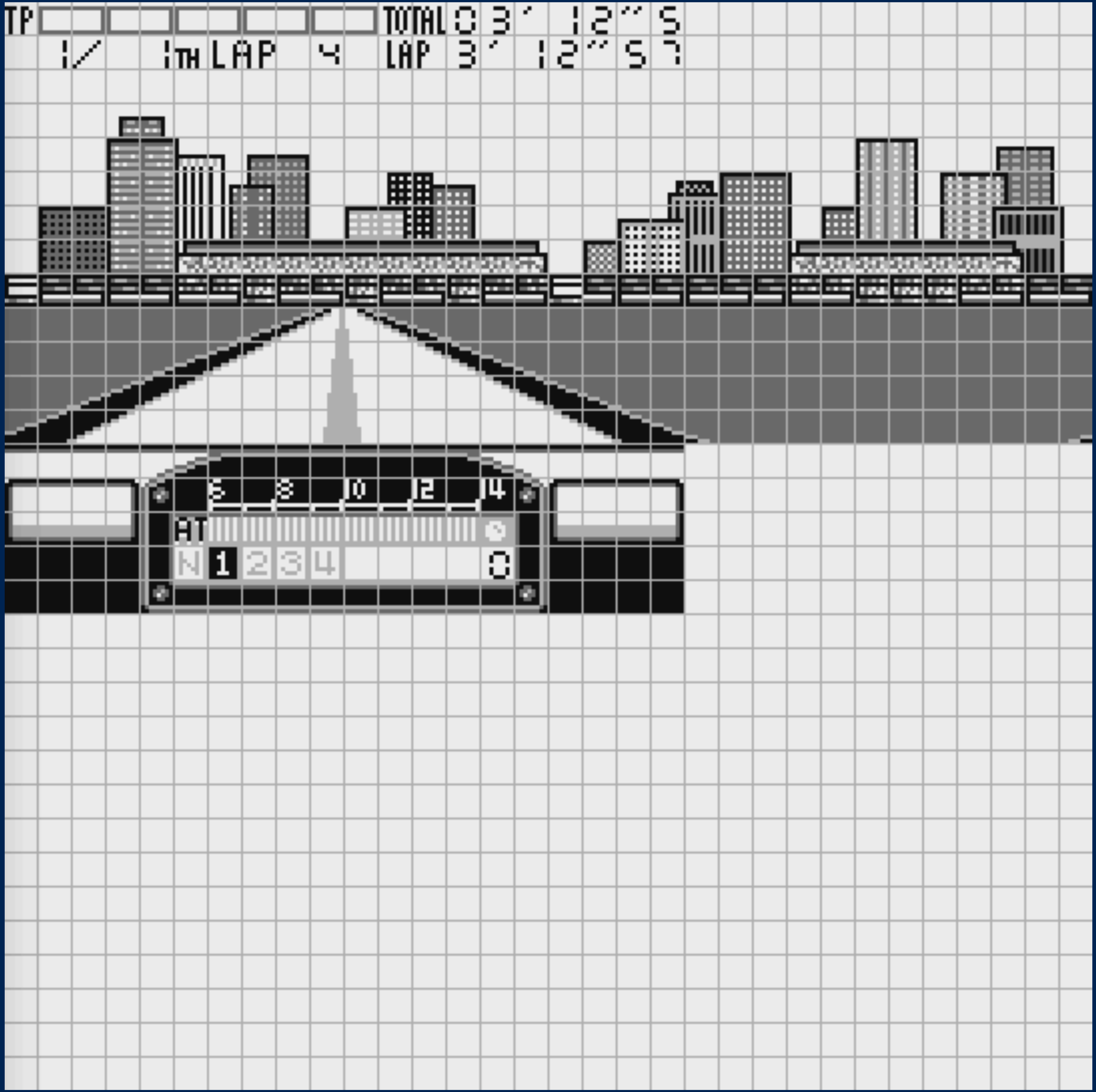
Changing SCX

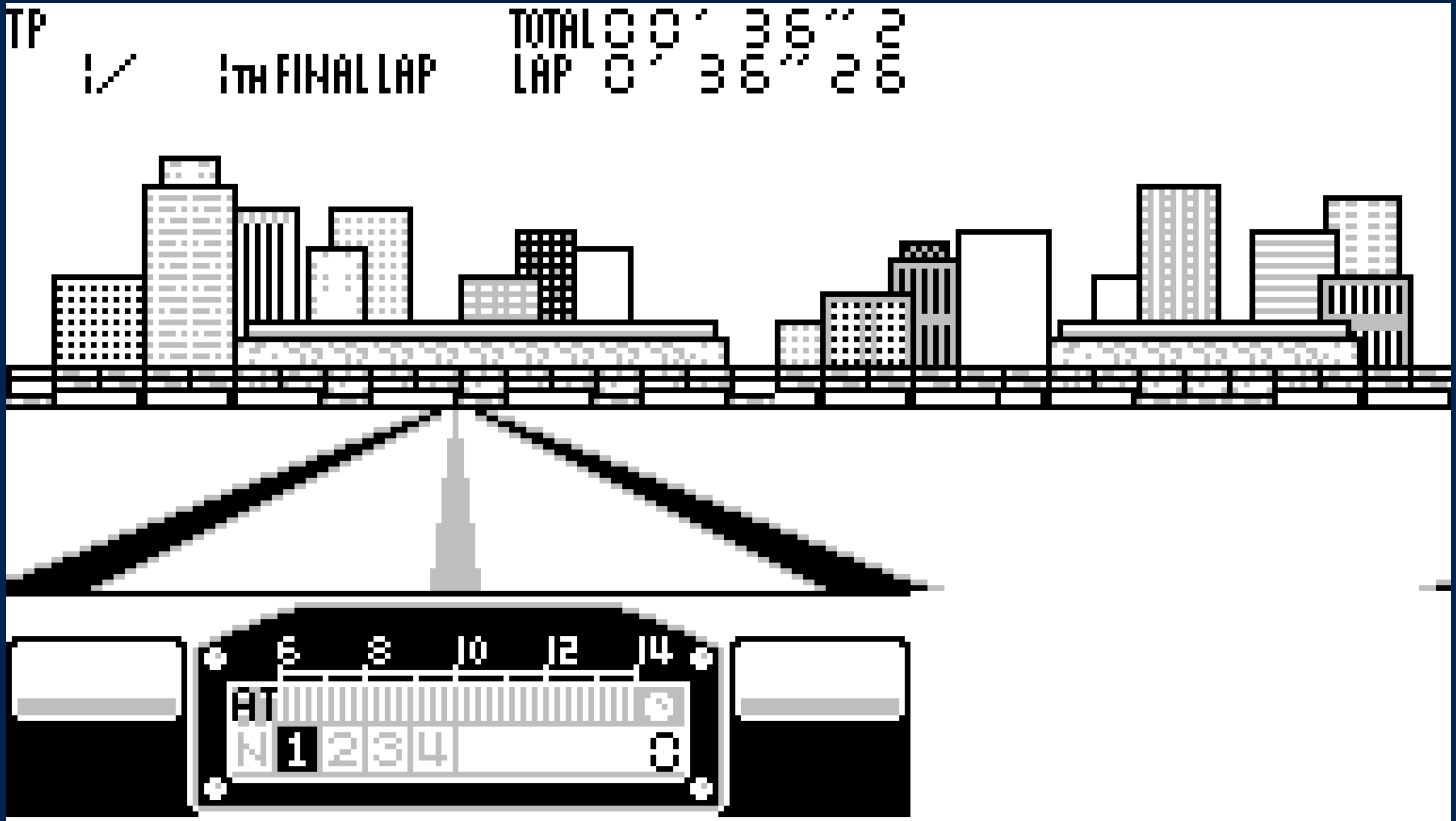


1/
1TH LAP 4
LAP 1′31″84
6 8 10 12 14
AT
N 1 2 3 4
54

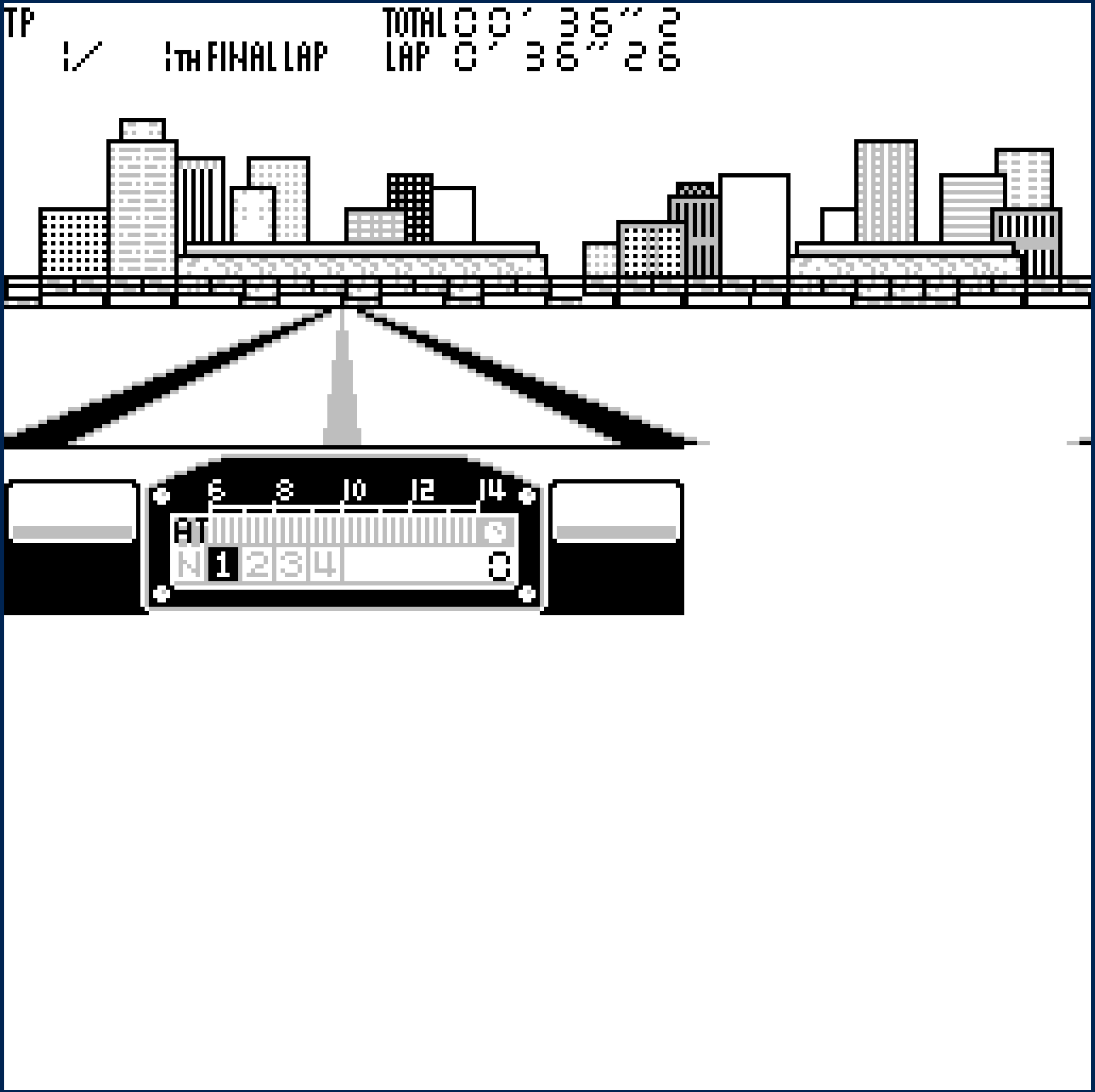

LAP 4
6 8 10 12 14
AT
N 1 2 3 4
54


1/
1TH LAP 4
LAP 1' 31"84
6 8 10 12 14
AT
N 1 2 3 4
54

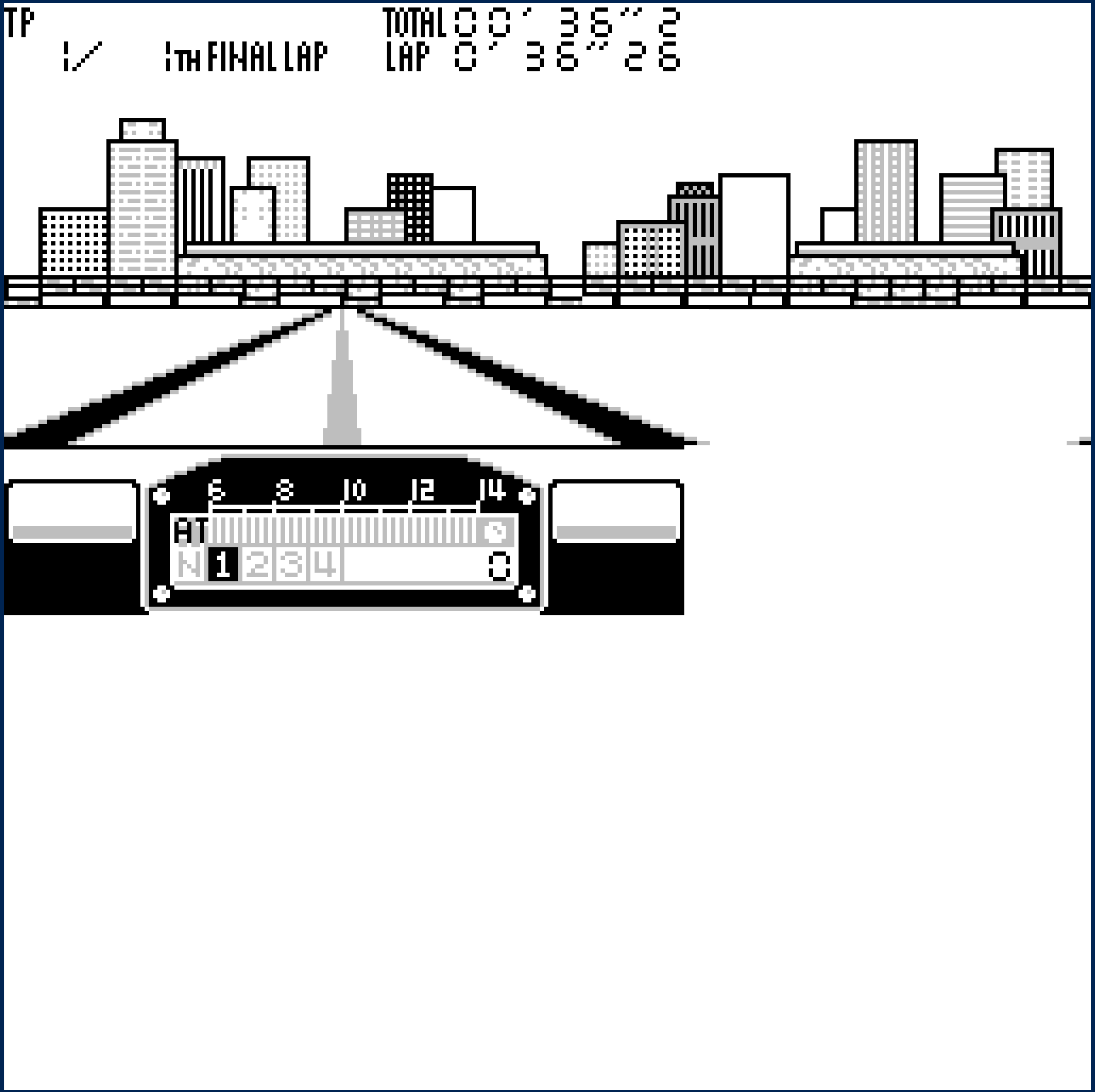

TP
TOTAL 03' 12" 5
1/
1TH LAP 4
LAP 3' 12" 57
6 8 10 12 14
AT
N 1 2 3 4
0


1/ 1TH LAP 4
LAP 1' 31"84
6 8 10 12 14
AT
N 1 2 3 4 54

TP
TOTAL 00' 36" 2
1/ 1TH FINAL LAP
LAP 0' 36" 26
6 8 10 12 14
AT
N 1 2 3 4 0

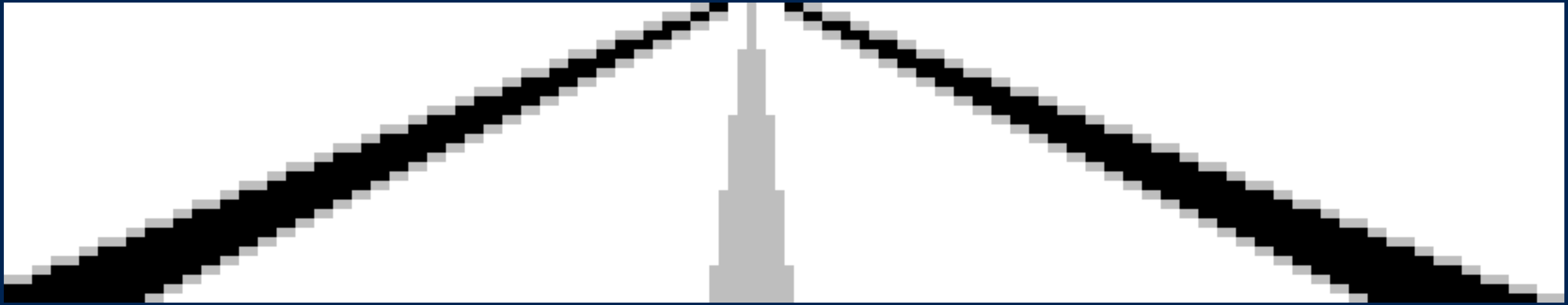
Changing SCX



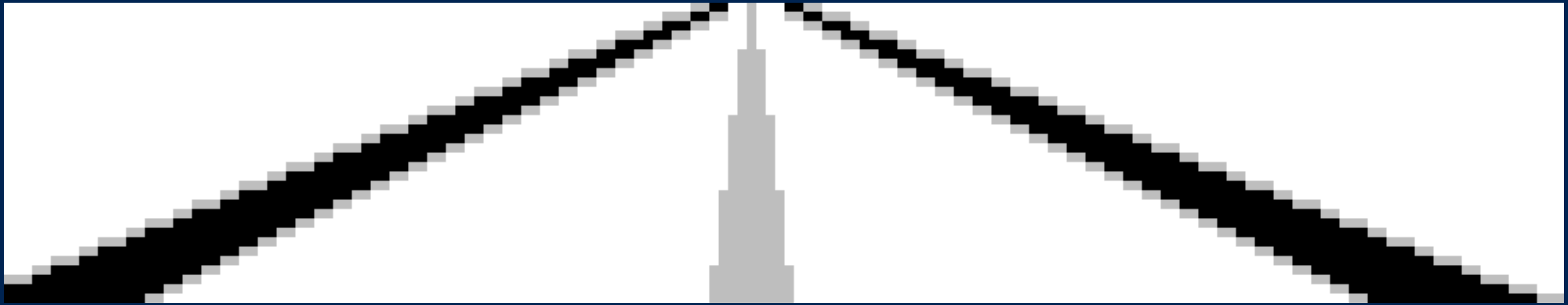
Changing SCX



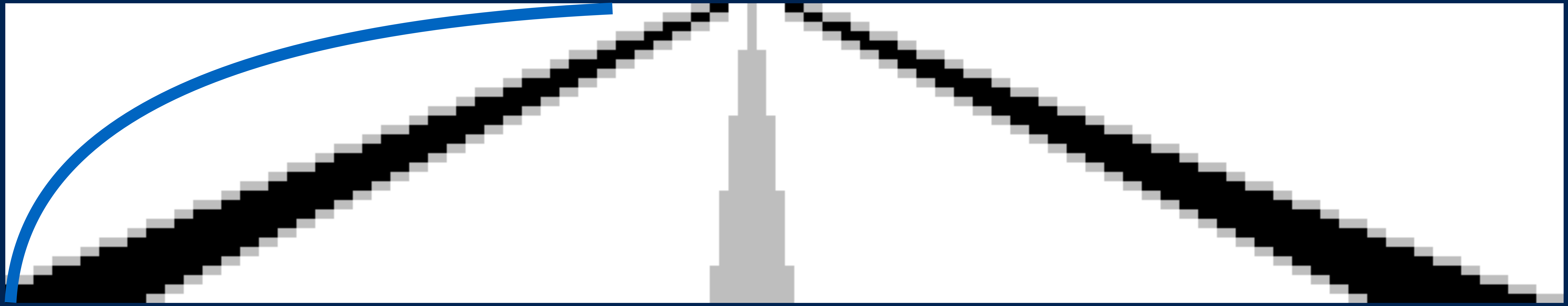
Changing SCX



Changing SCX



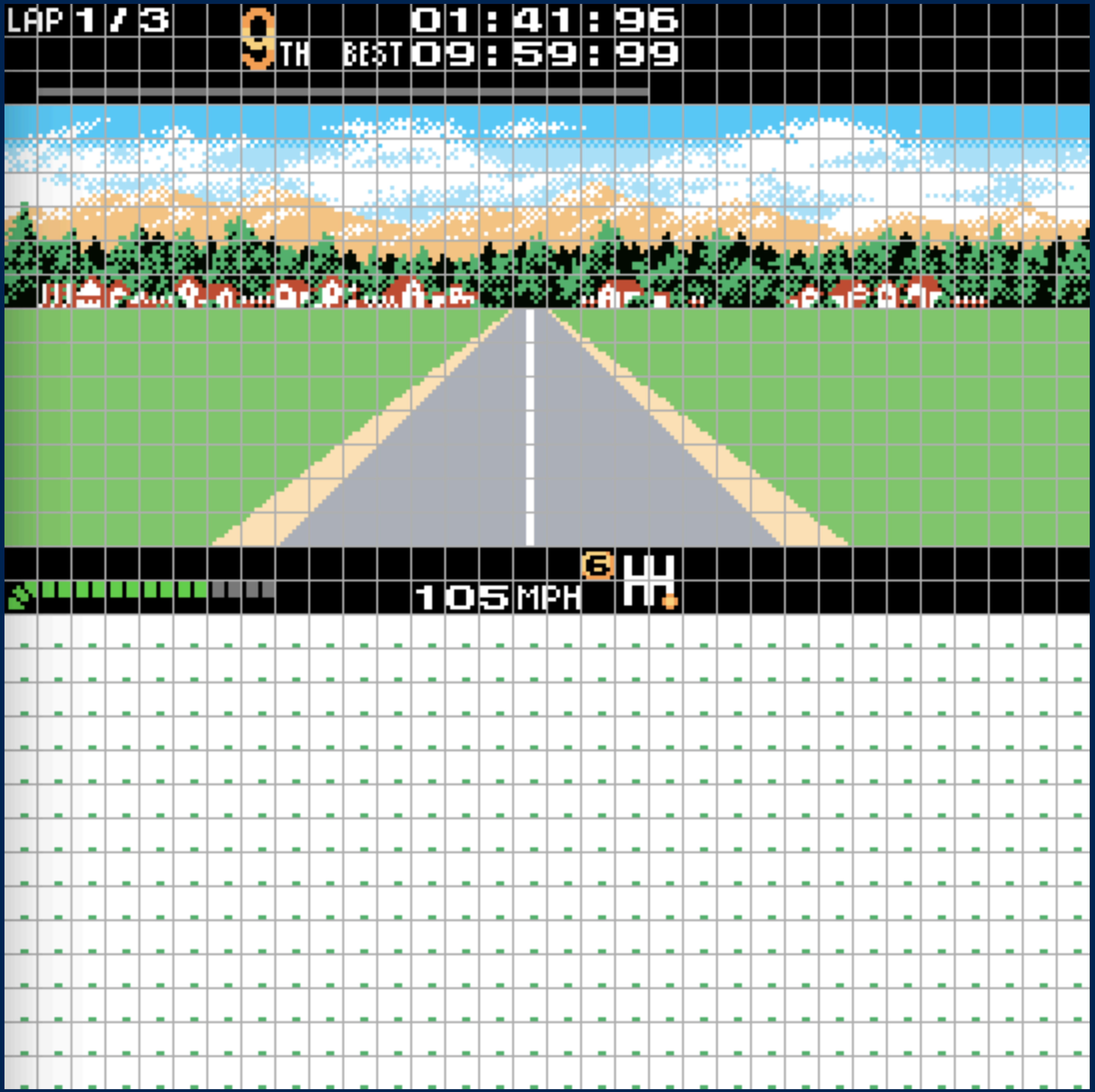
Changing SCX



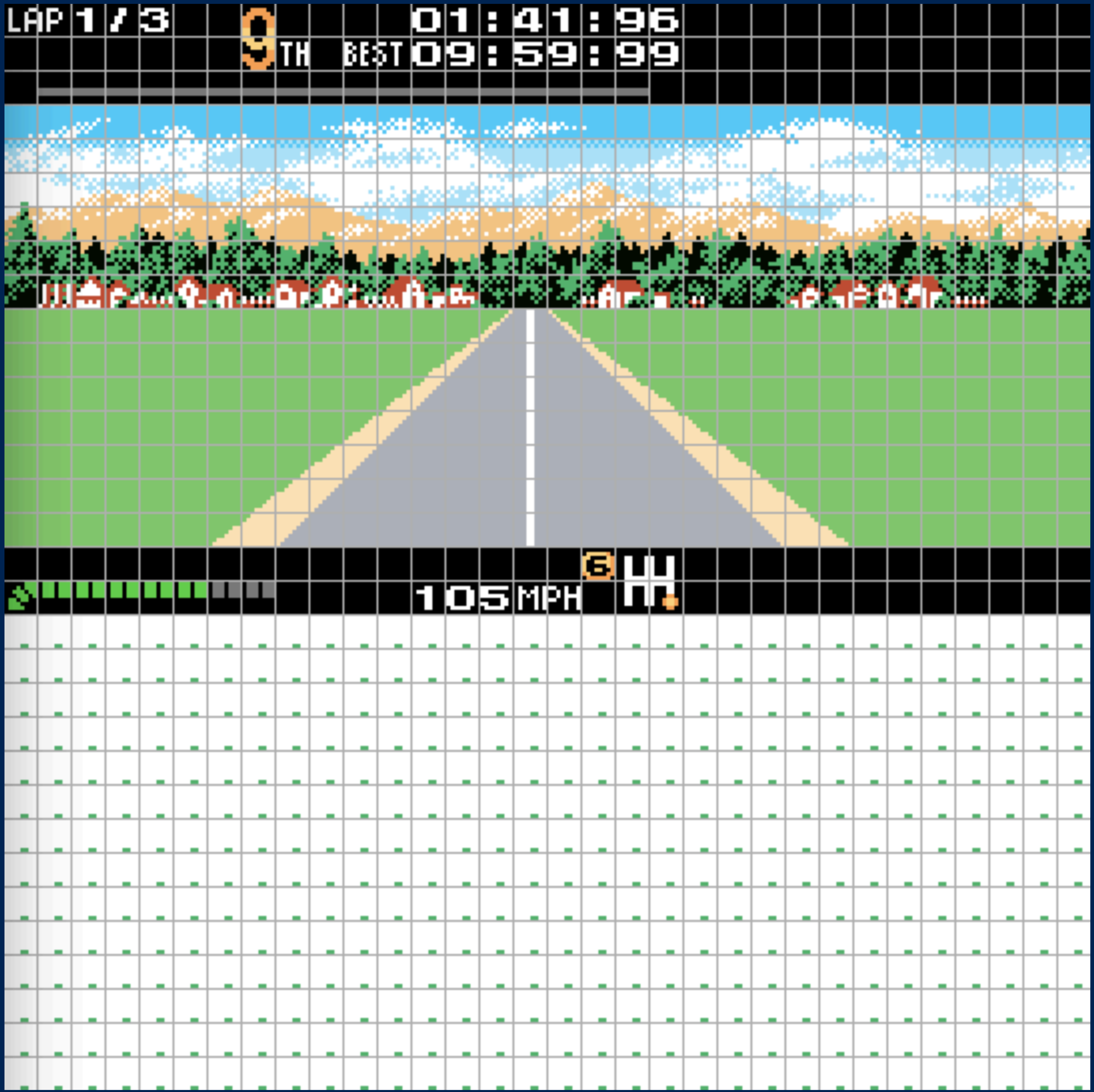
Changing SCX

Changing SCX


LAP 1/3
9TH
01:31:71
BEST 09:59:99
6
135 MPH


LAP 1/3
9TH
01:41:96
BEST 09:59:99
6
105 MPH


LAP 1/3
9TH
01:31:71
BEST 09:59:99
6
135 MPH


LAP 1/3
9TH
01:41:96
BEST 09:59:99
6
105 MPH


DOT MATRIX WITH STEREO SOUND
BATTERY


BATTERY

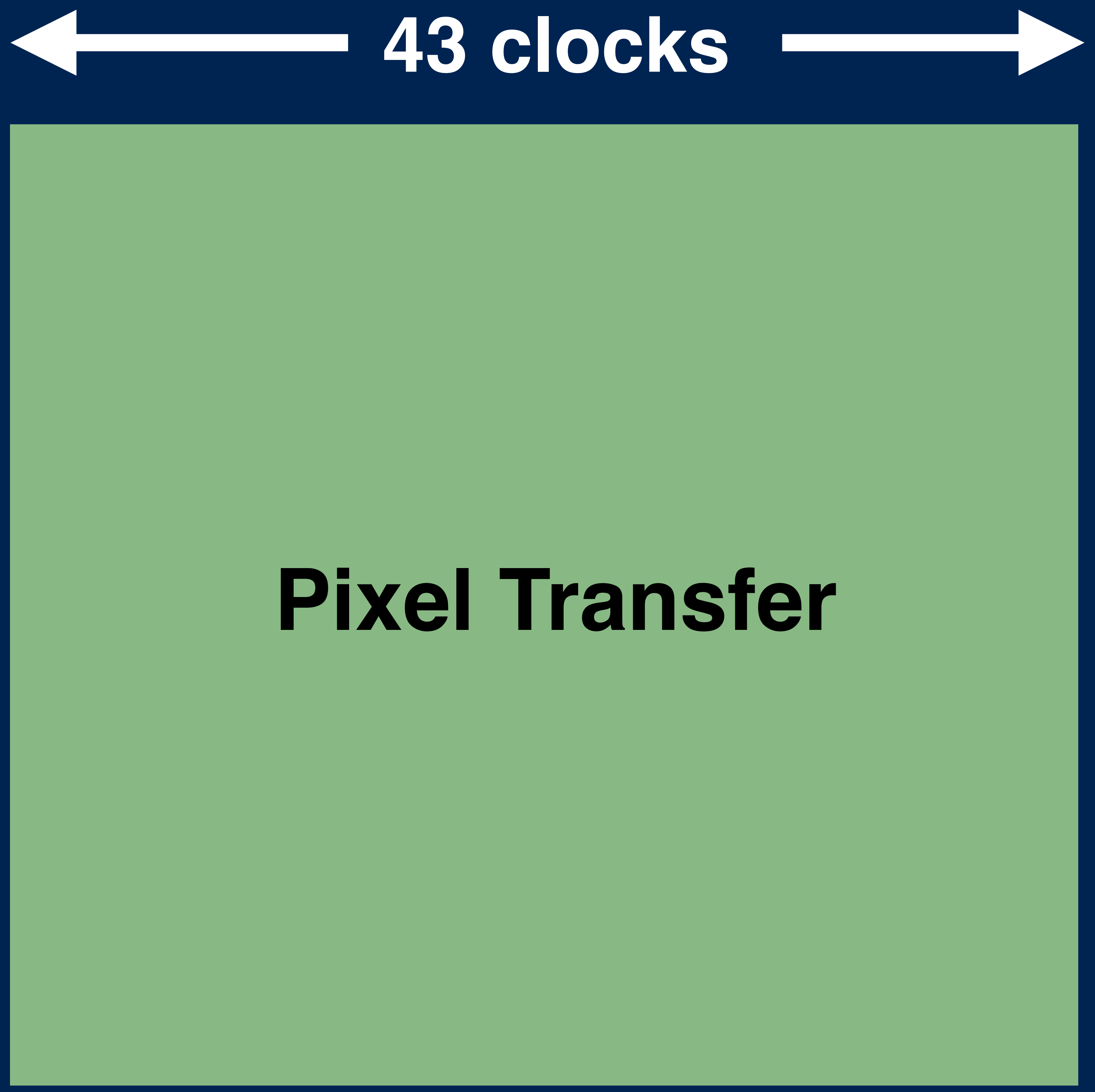
Pixel Processing Unit

Pixel Processing Unit

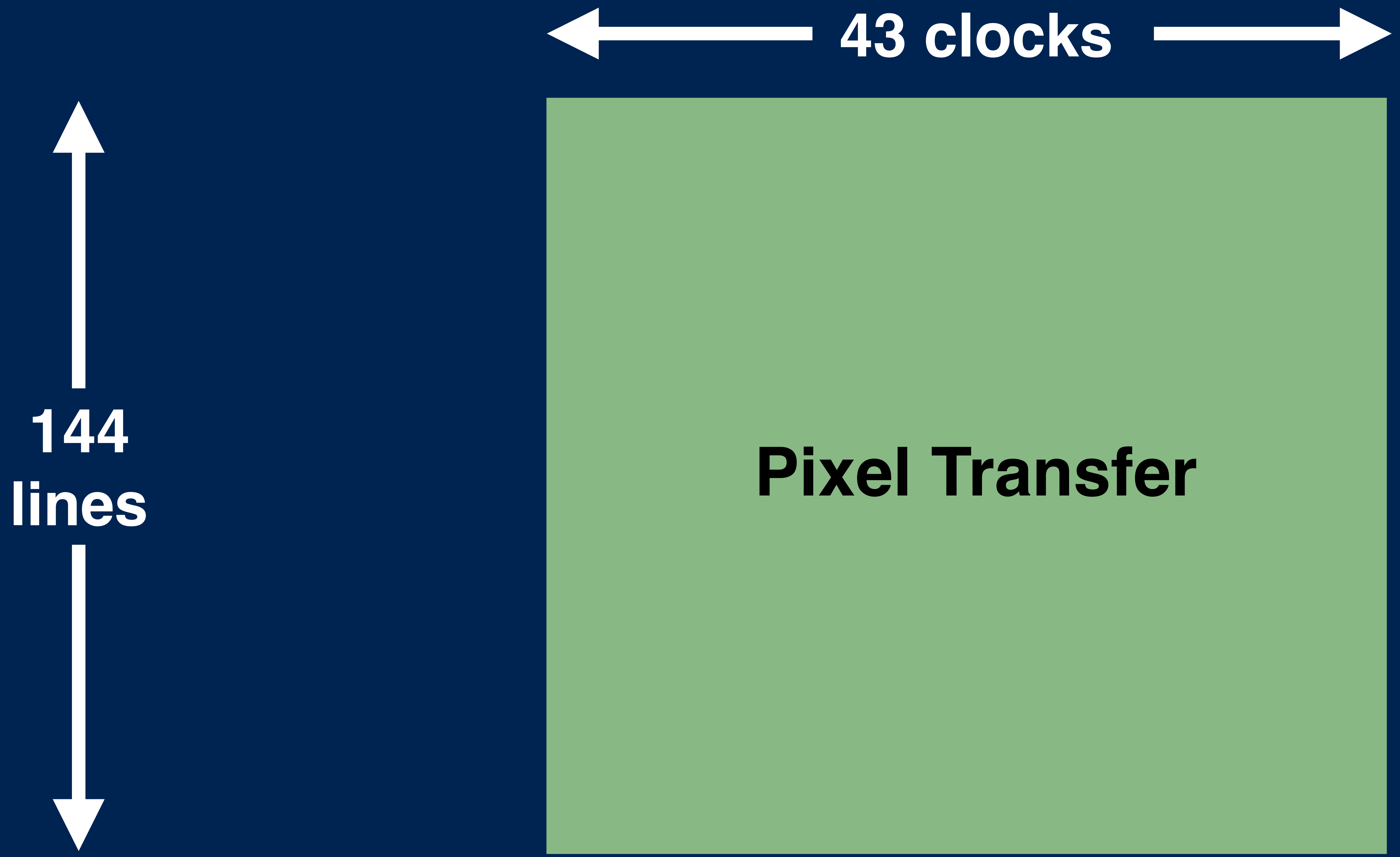
Horizontal Timing


DOT MATRIX WITH STEREO SOUND
BATTERY
Nintendo GAME BOY TM

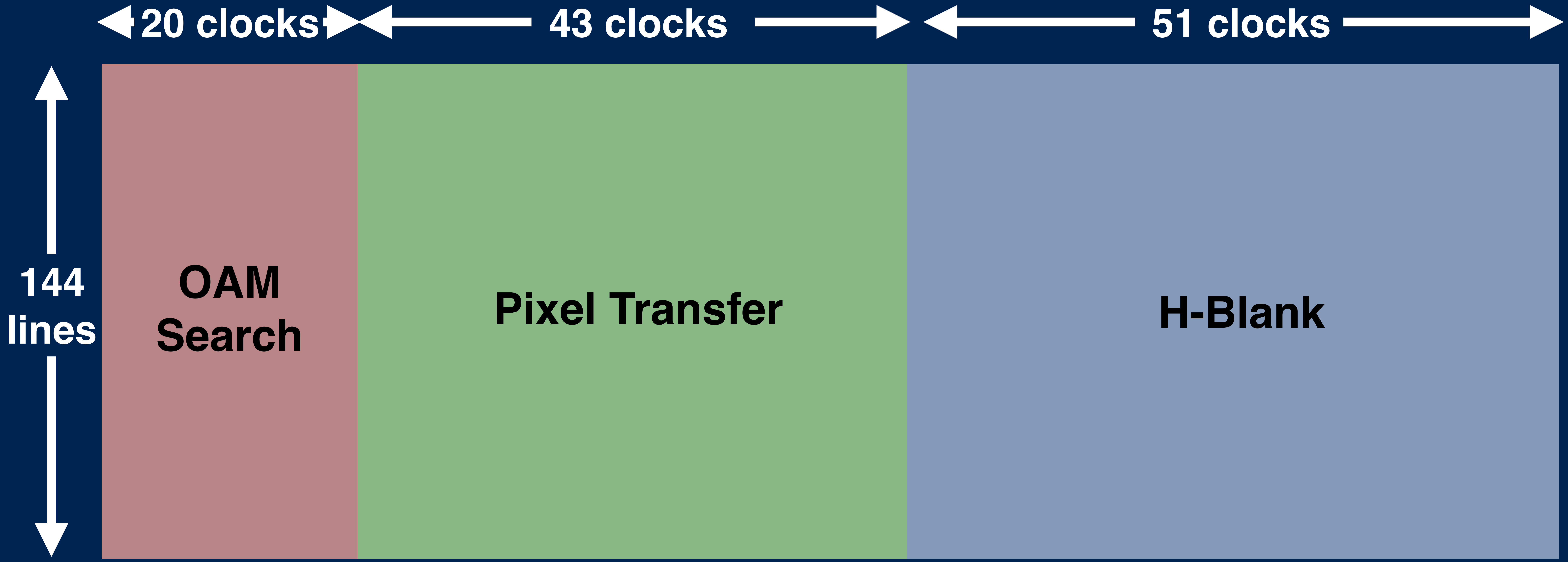
PPU Timing



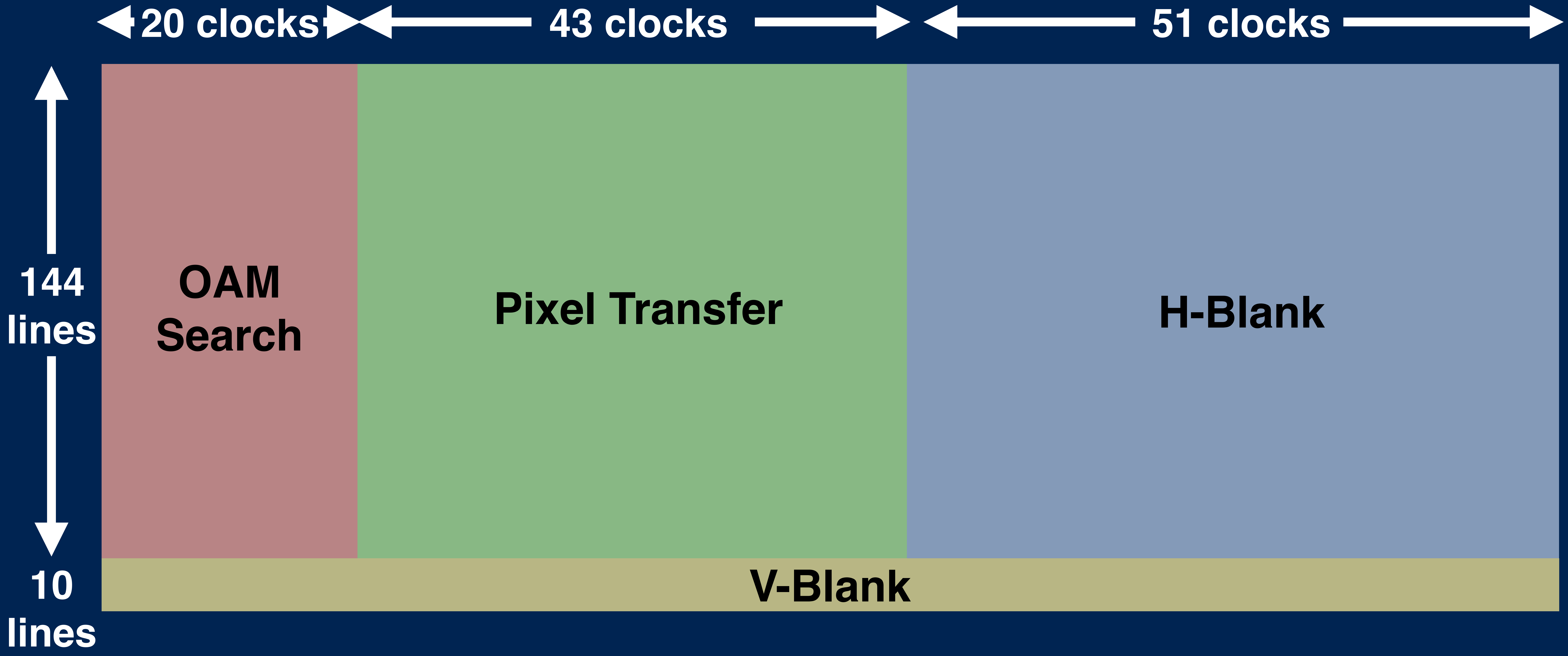
PPU Timing



PPU Timing

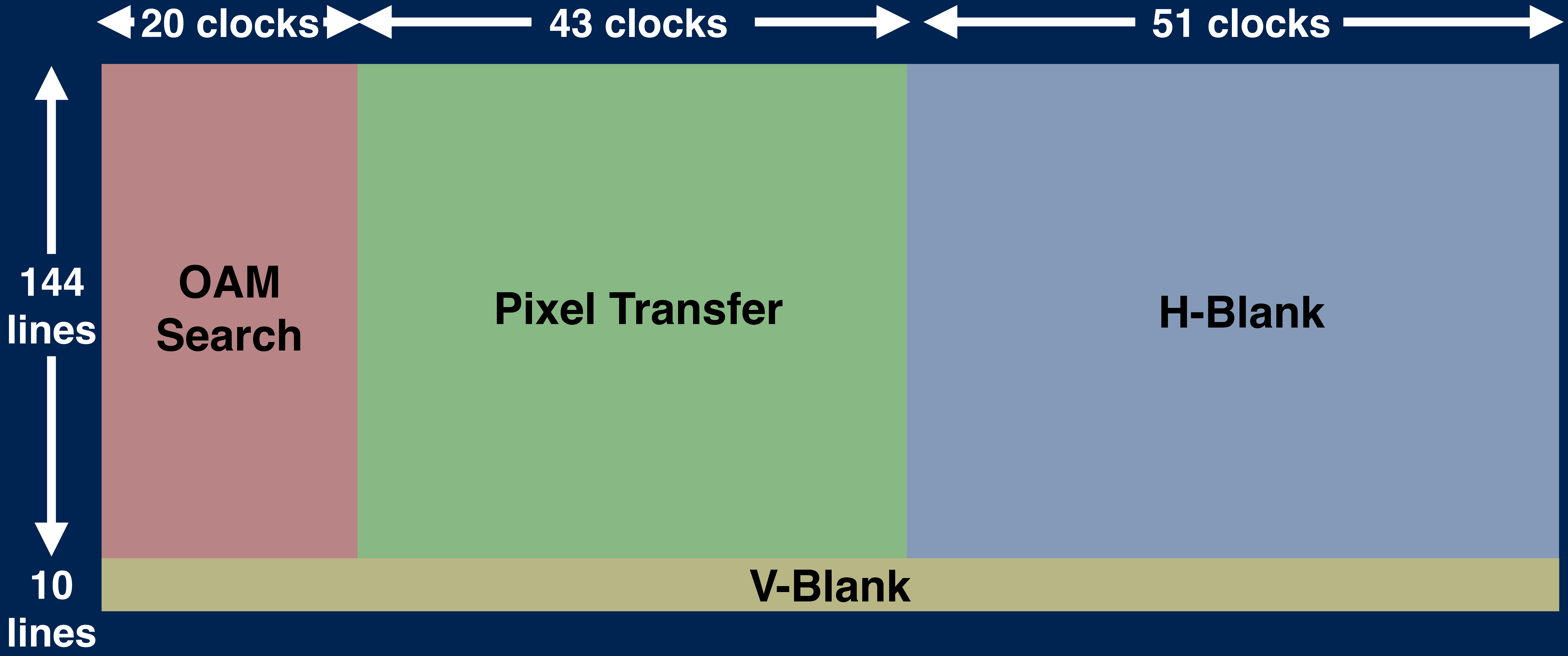

20 clocks
43 clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank

PPU Timing



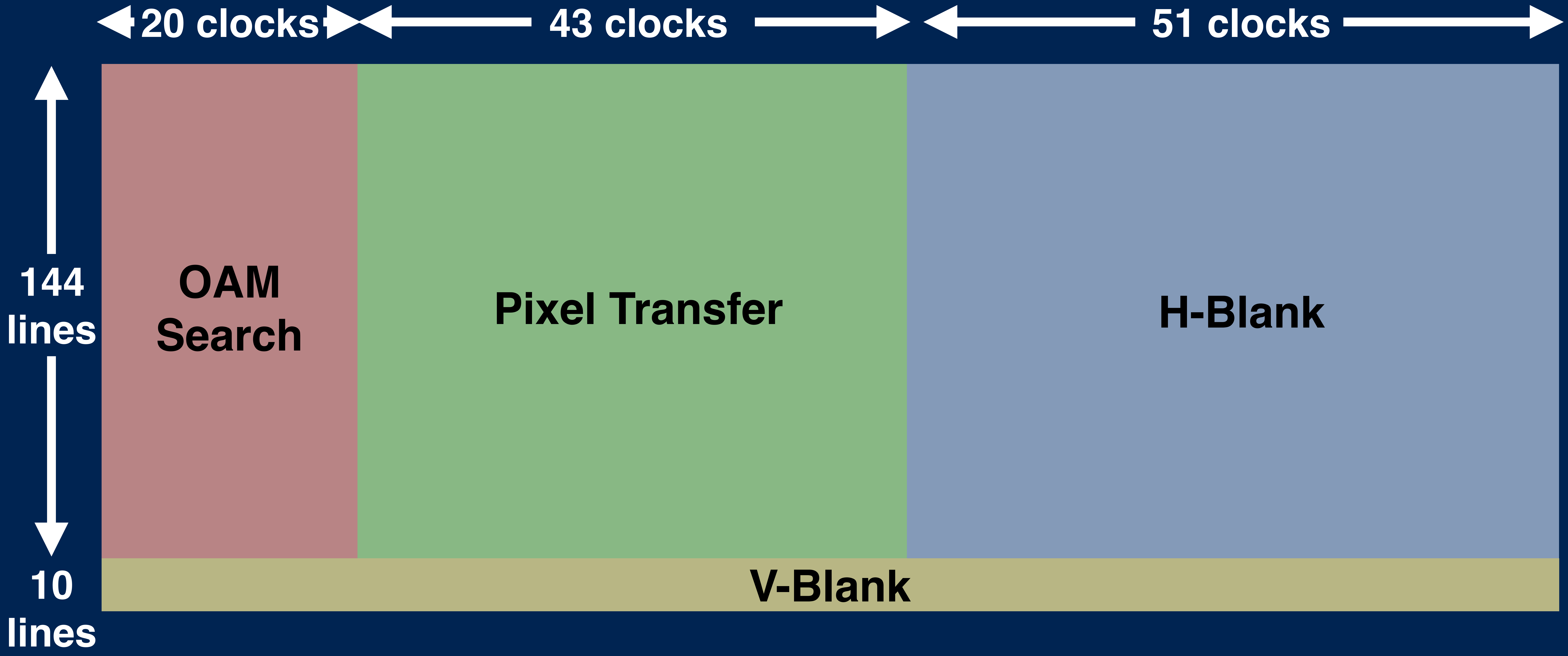
20 clocks
43 clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
10 lines
V-Blank

PPU Timing



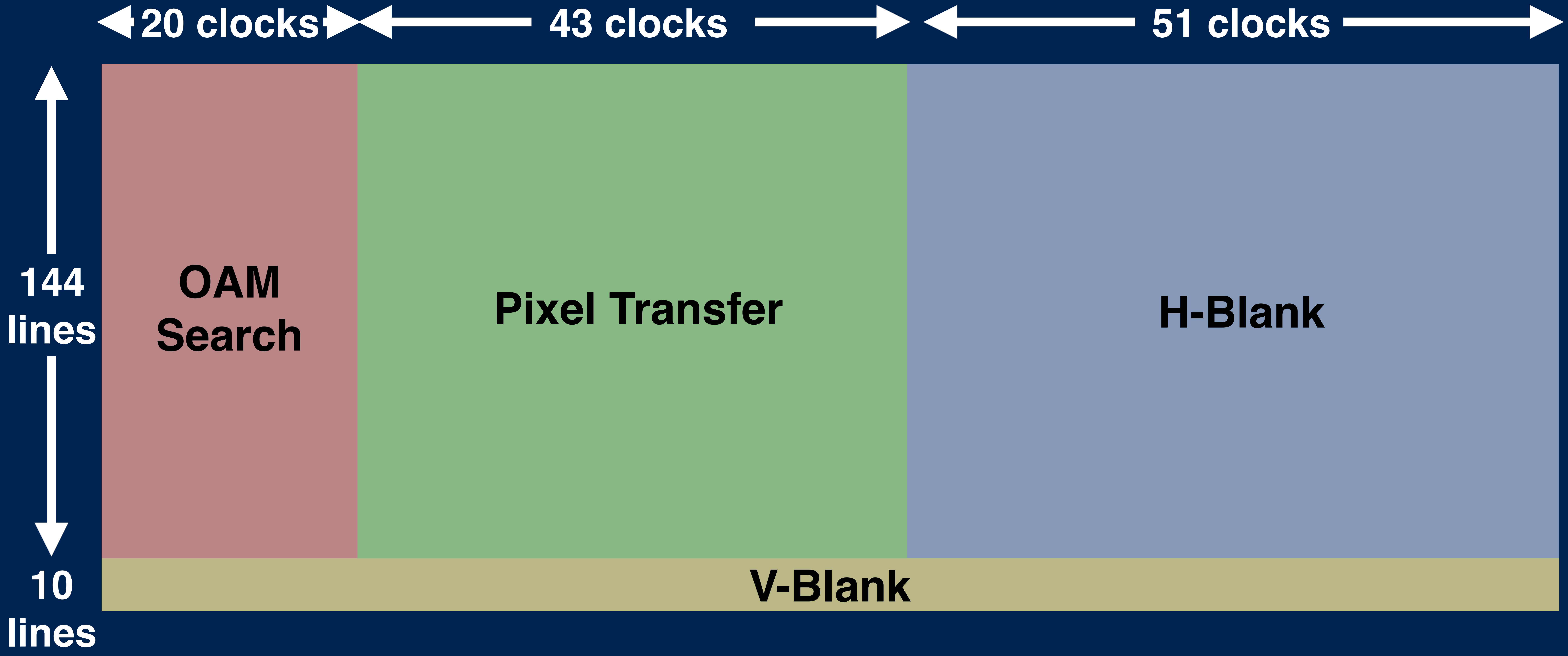
114 x 154

PPU Timing



114 x 154 = 17,556 clocks per screen

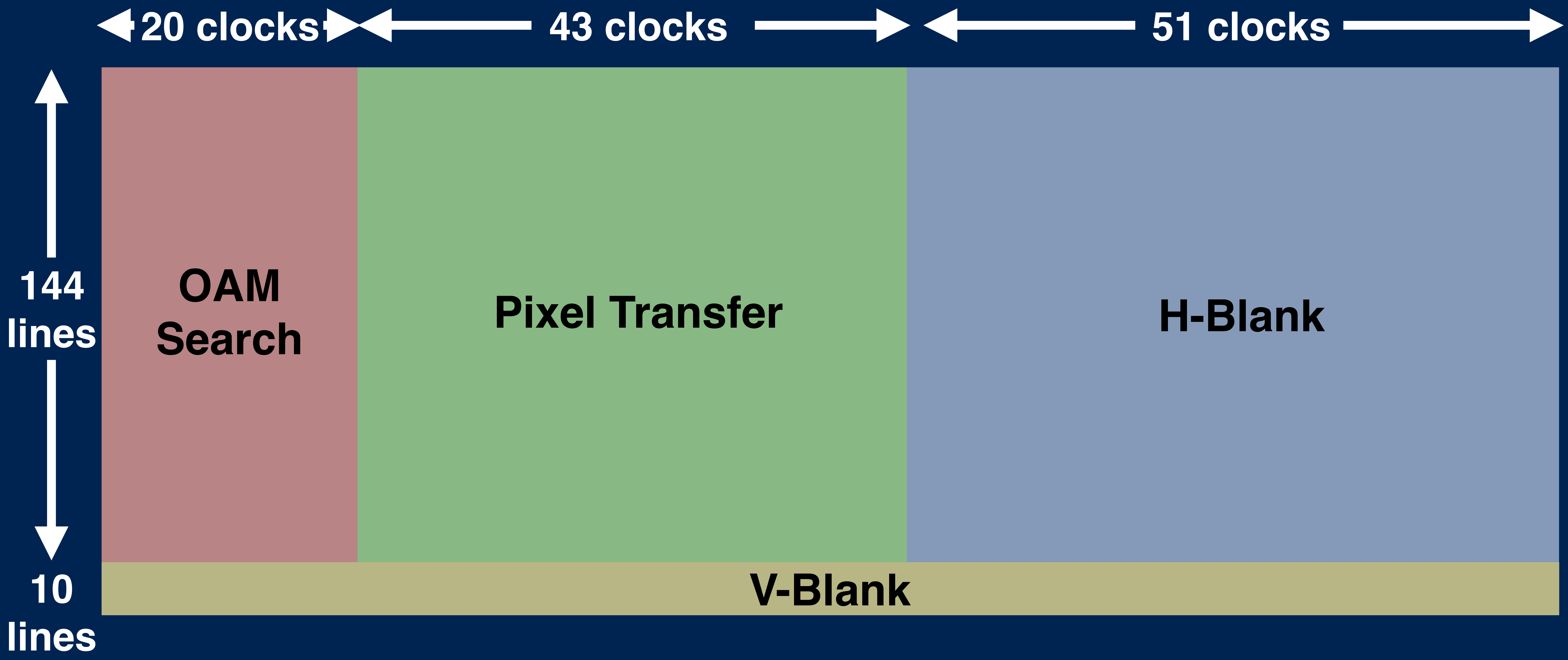
PPU Timing



114 x 154 = 17,556 clocks per screen

1,048,576 / 17,556

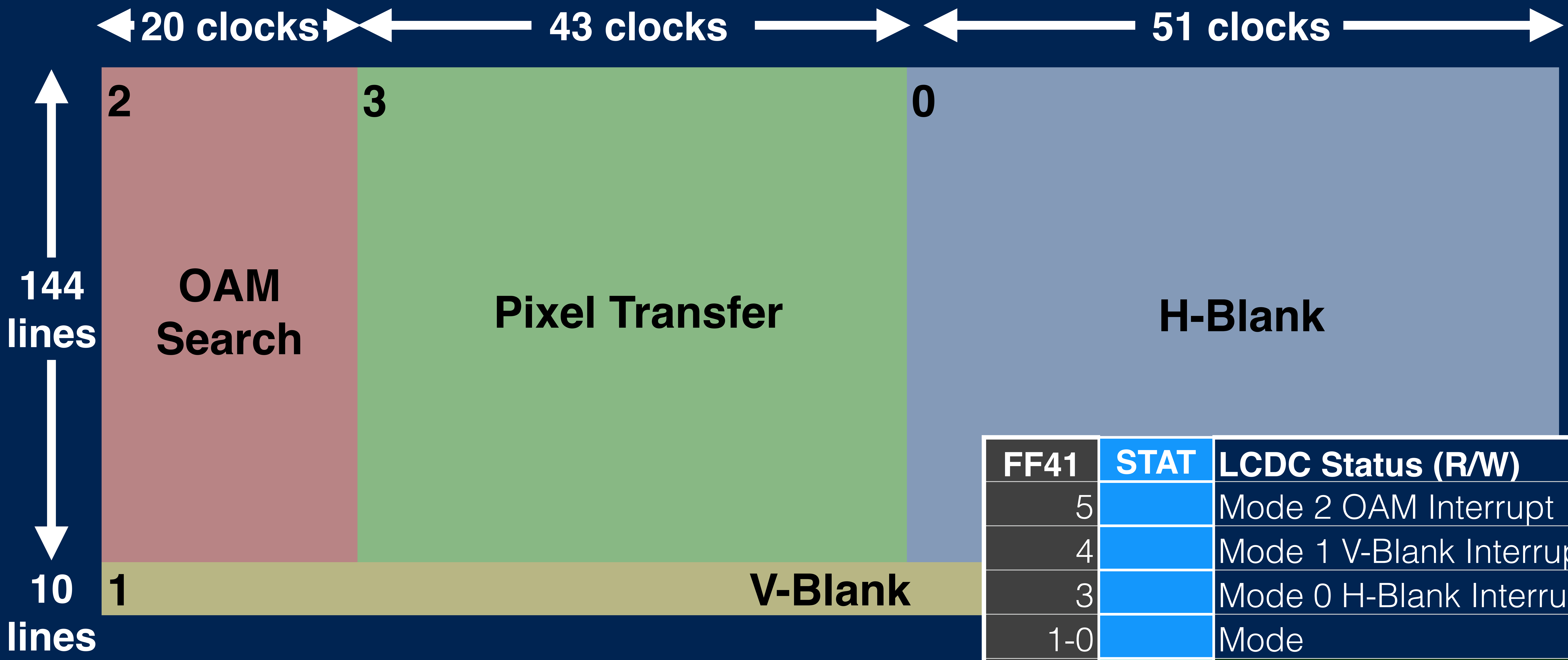
PPU Timing



114 x 154 = 17,556 clocks per screen

1,048,576 / 17,556 = 59.7 Hz refresh rate

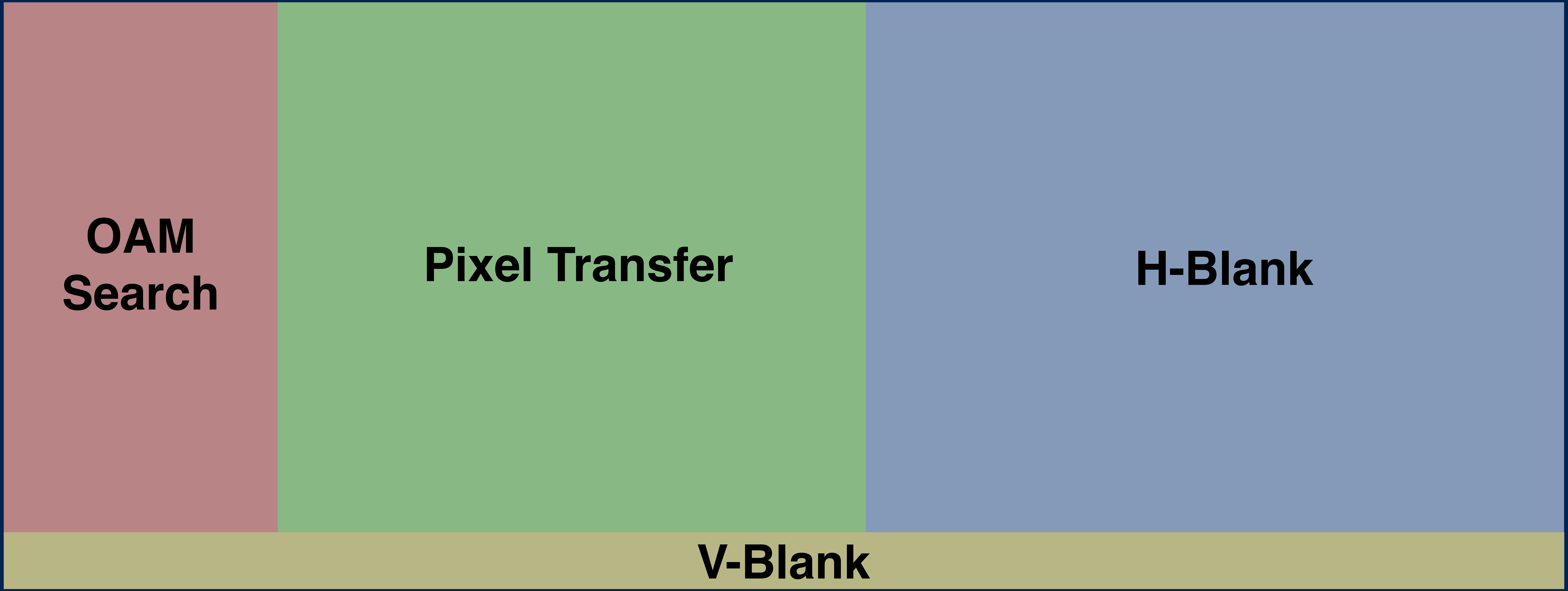
PPU Timing



FF41	**STAT**	**LCDC Status (R/W)**
5		Mode 2 OAM Interrupt
4		Mode 1 V-Blank Interrupt
3		Mode 0 H-Blank Interrupt
1-0		Mode
FF0F	**IF**	**Interrupt Flag (R/W)**
0		V-Blank
FFFF	**IE**	**Interrupt Enable (R/W)**
0		V-Blank

114 x 154 = 17,556 clocks per screen

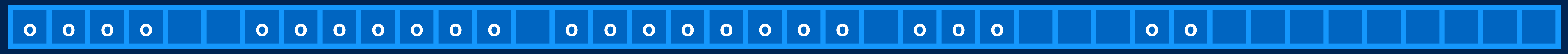
1,048,576 / 17,556 = 59.7 Hz refresh rate


OAM
Search
Pixel Transfer
H-Blank
V-Blank

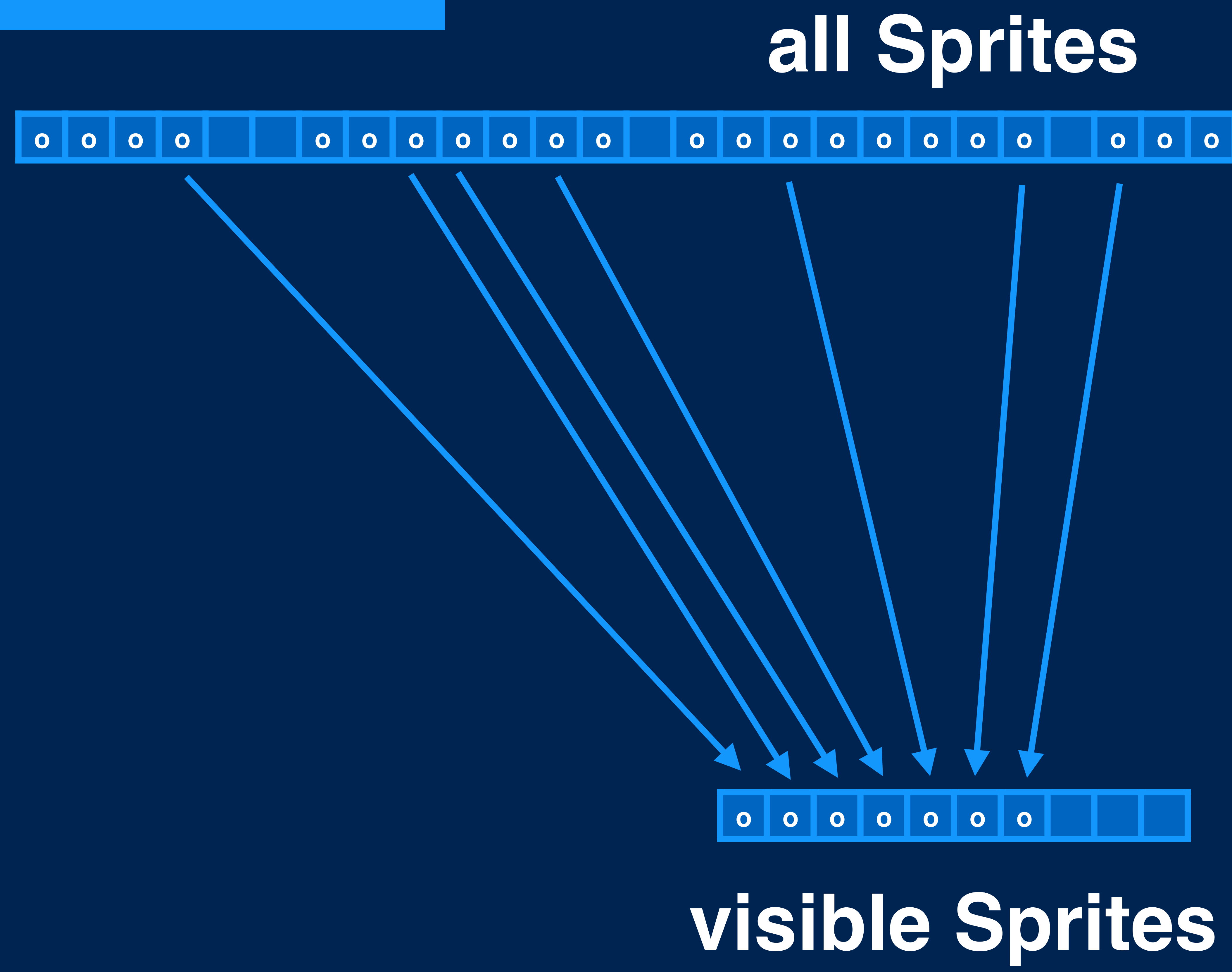
OAM
Search

OAM Search

all Sprites



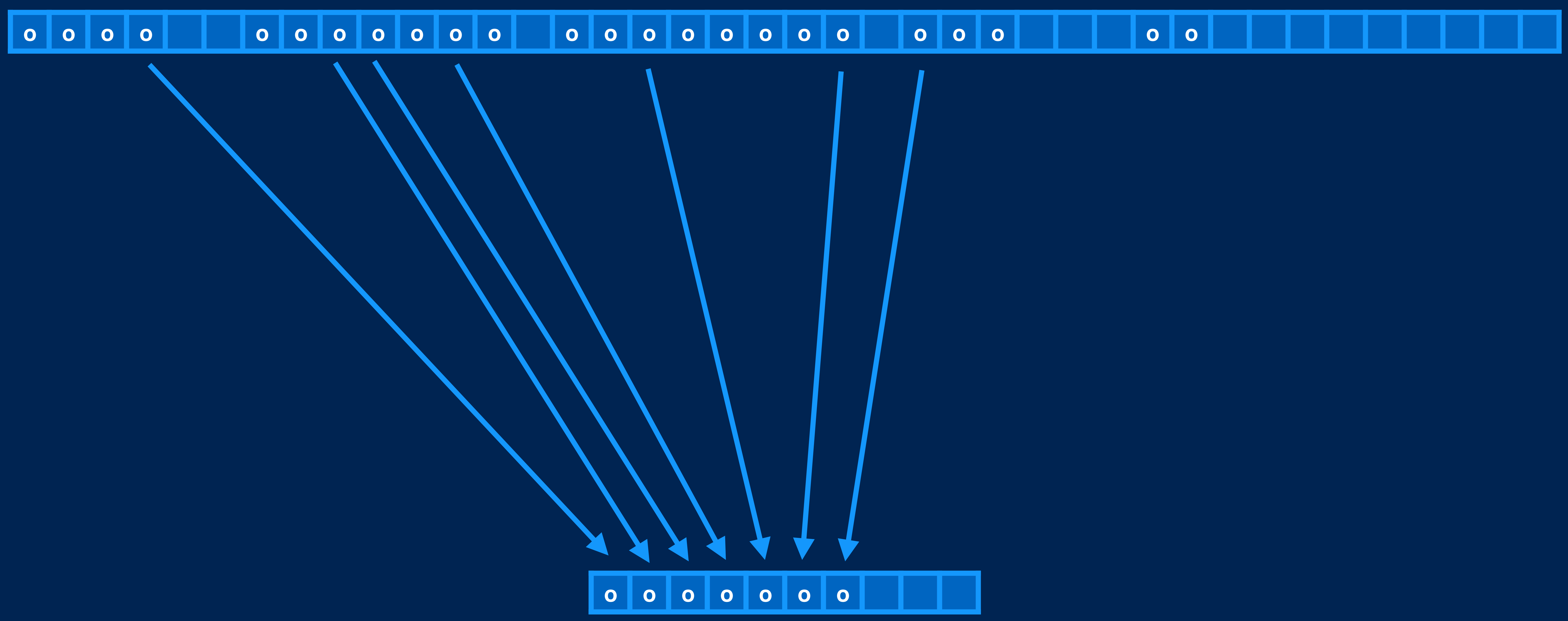
OAM Search


all Sprites
o o o o o o o o o o o o o o o o o o o o o o
o o o o o o o
visible Sprites


OAM Search

OAM Search

all Sprites



visible Sprites

$$
\begin{aligned}
& \mathtt{oam.x} \neq 0 \\
\wedge\ & \mathtt{LY} + 16 \geq \mathtt{oam.y} \\
\wedge\ & \mathtt{LY} + 16 < \mathtt{oam.y} + \mathtt{h}
\end{aligned}
$$

OAM Search CPU Bug

OAM Search CPU Bug

r16 inc/dec **FE00 ≤ r16 ≤ FEFF**

```
inc bc      dec bc      ld a, (hl+)
inc de      dec de      ld a, (hl-)
inc hl      dec hl      ld (hl+), a
inc sp      dec sp      ld (hl-), a
```

during OAM search

OAM Search CPU Bug

r16 inc/dec **FE00 ≤ r16 ≤ FEFF**

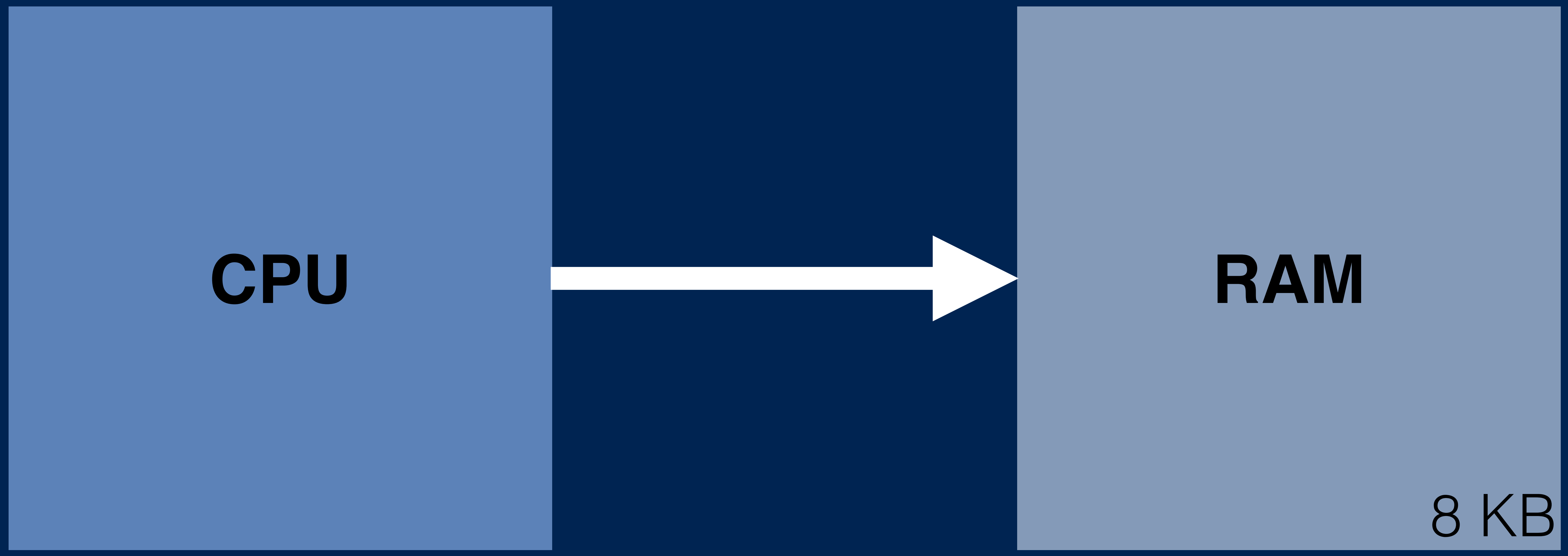
```
inc bc      dec bc      ld a, (hl+)
inc de      dec de      ld a, (hl-)
inc hl      dec hl      ld (hl+), a
inc sp      dec sp      ld (hl-), a
```

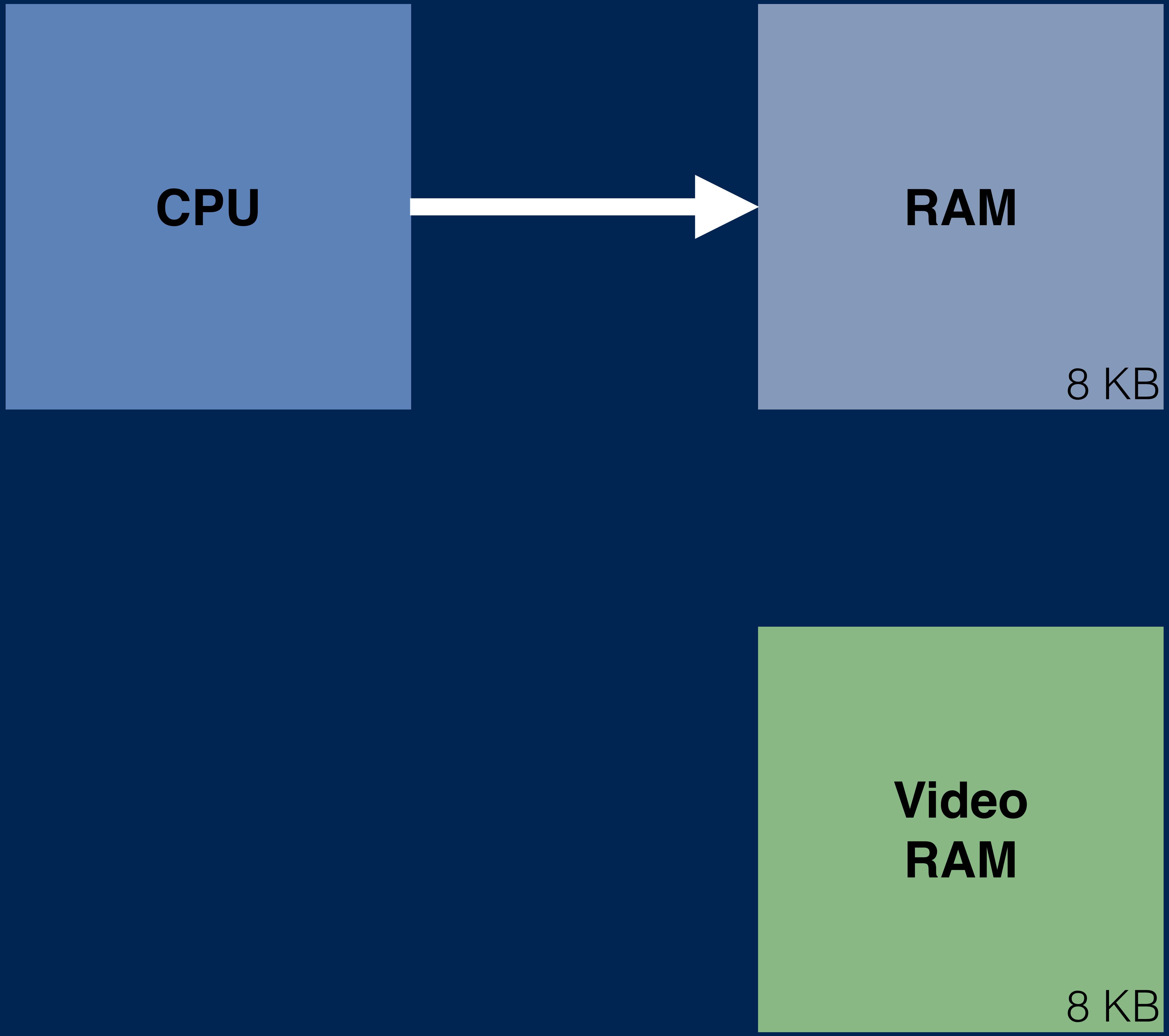
during OAM search

will trash OAM RAM

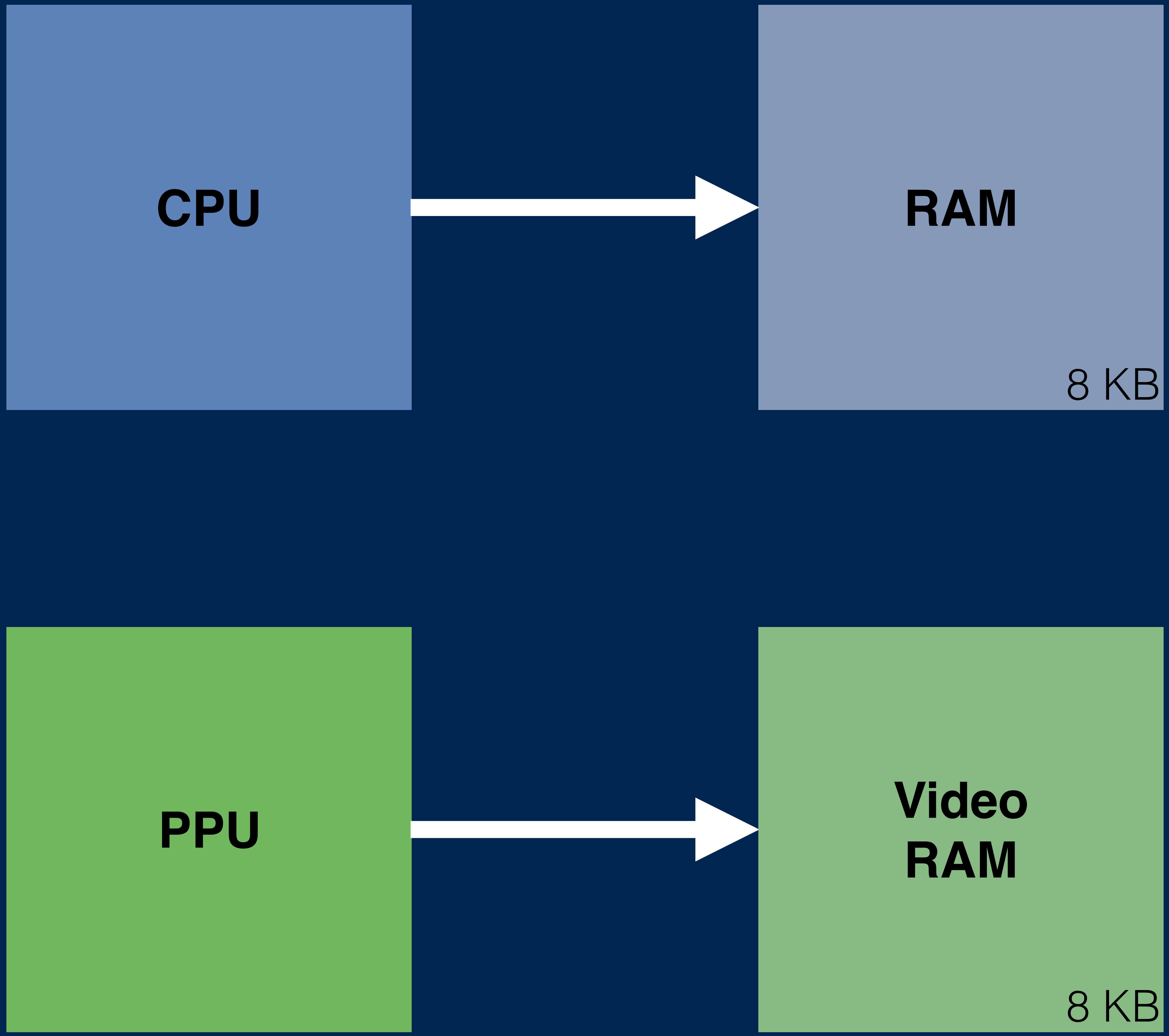
RAM

8 KB

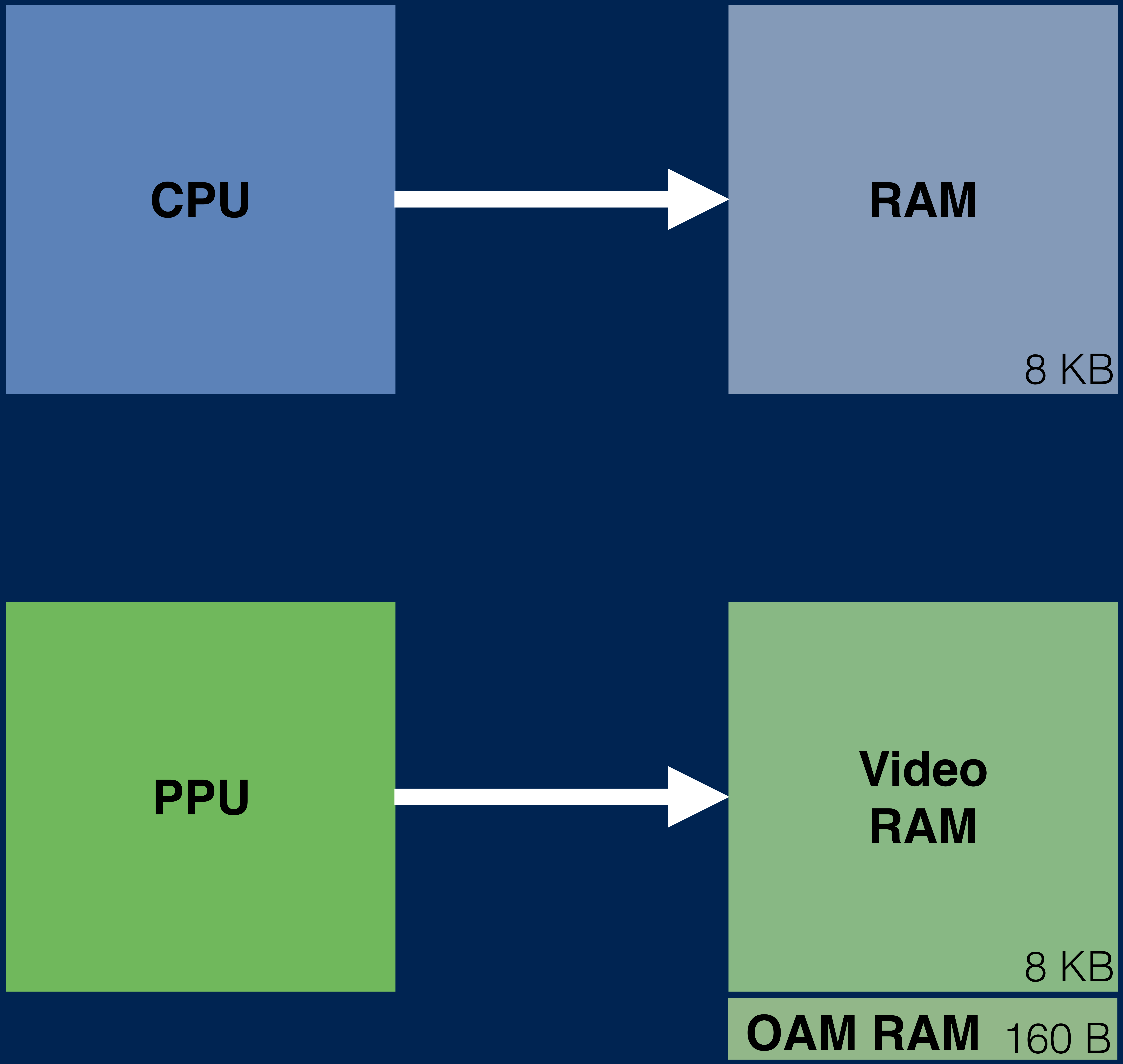

CPU
RAM
8 KB


CPU
RAM
8 KB
Video
RAM
8 KB

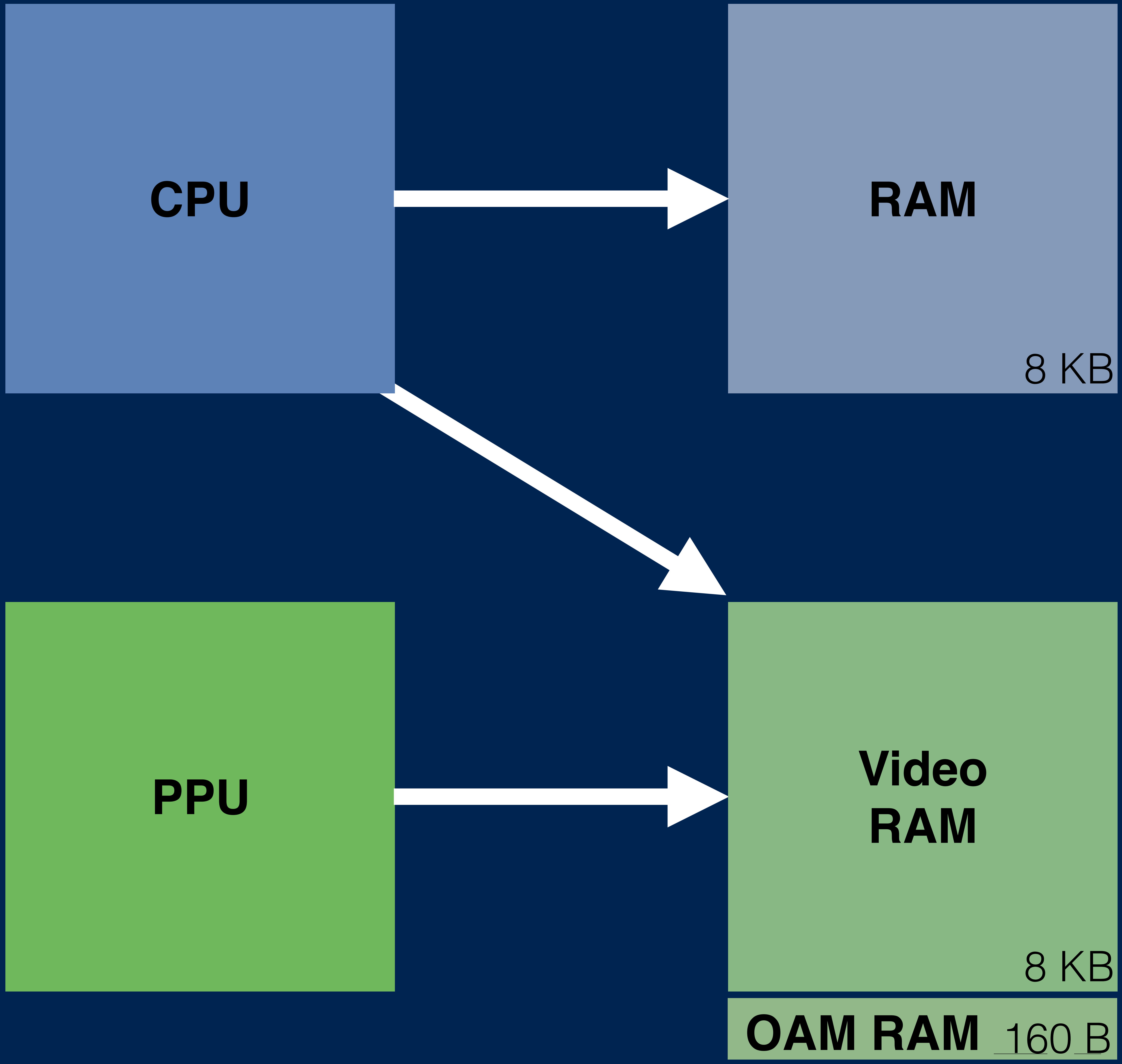
VRAM Access



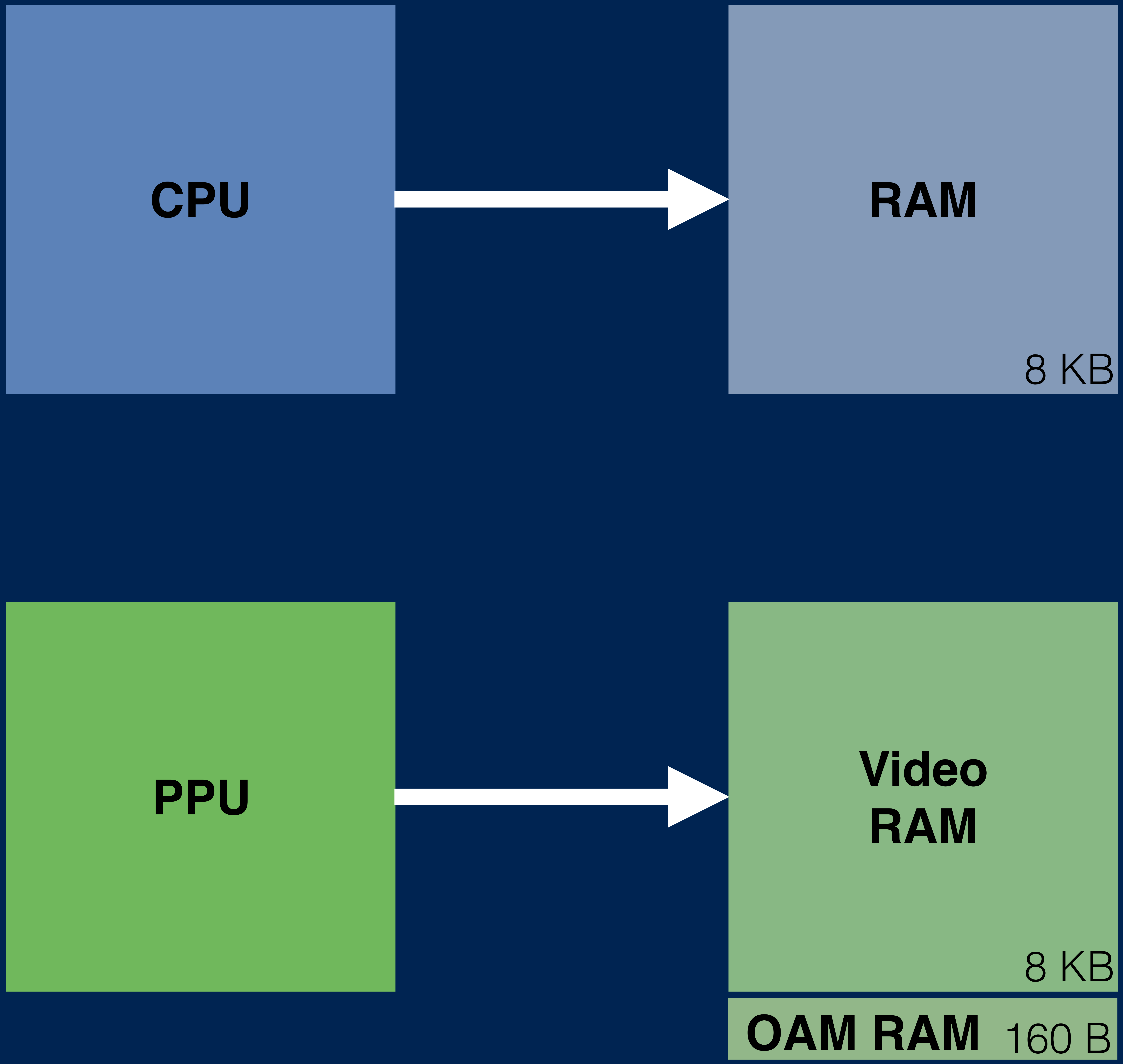
VRAM Access



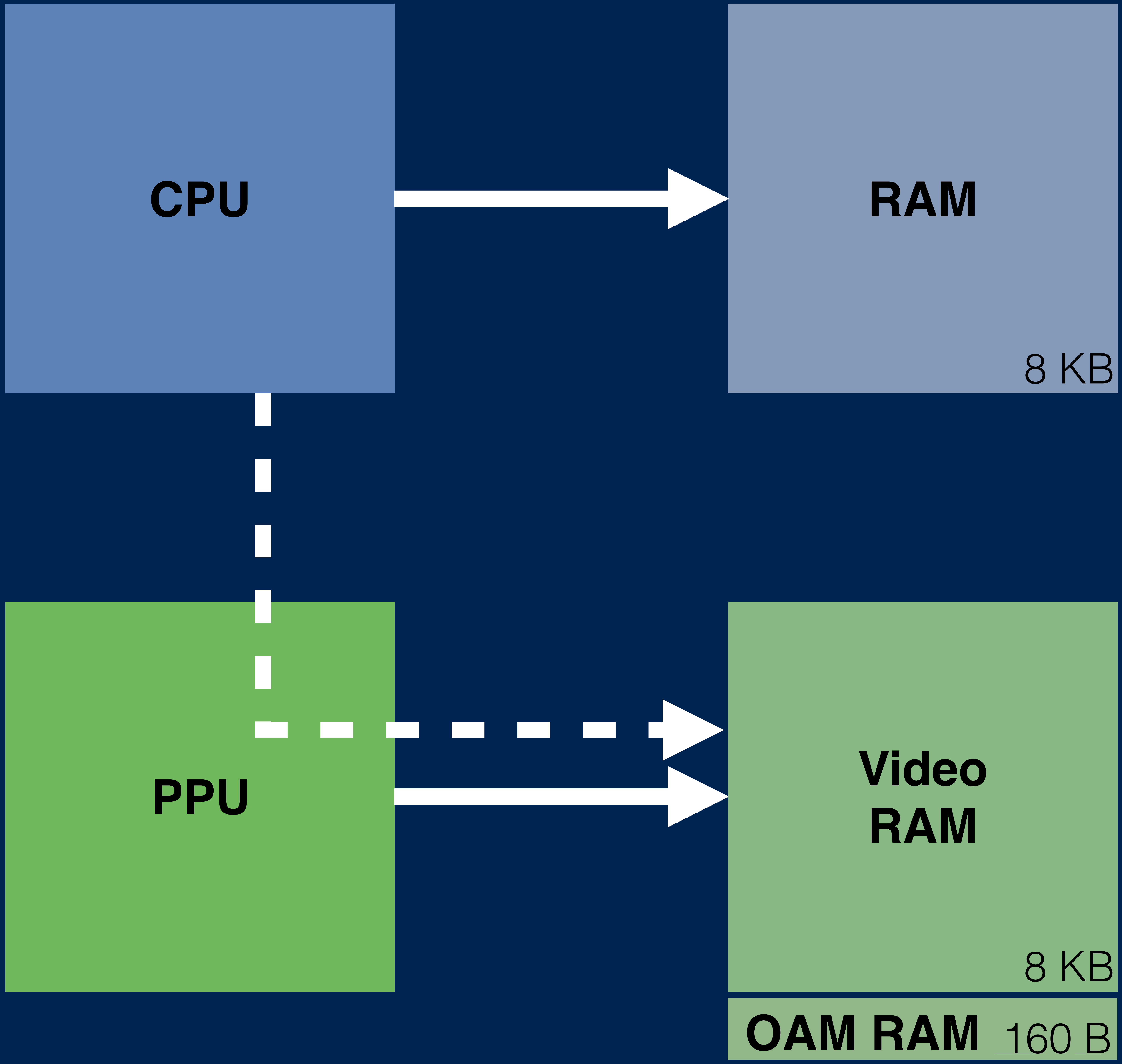
VRAM Access



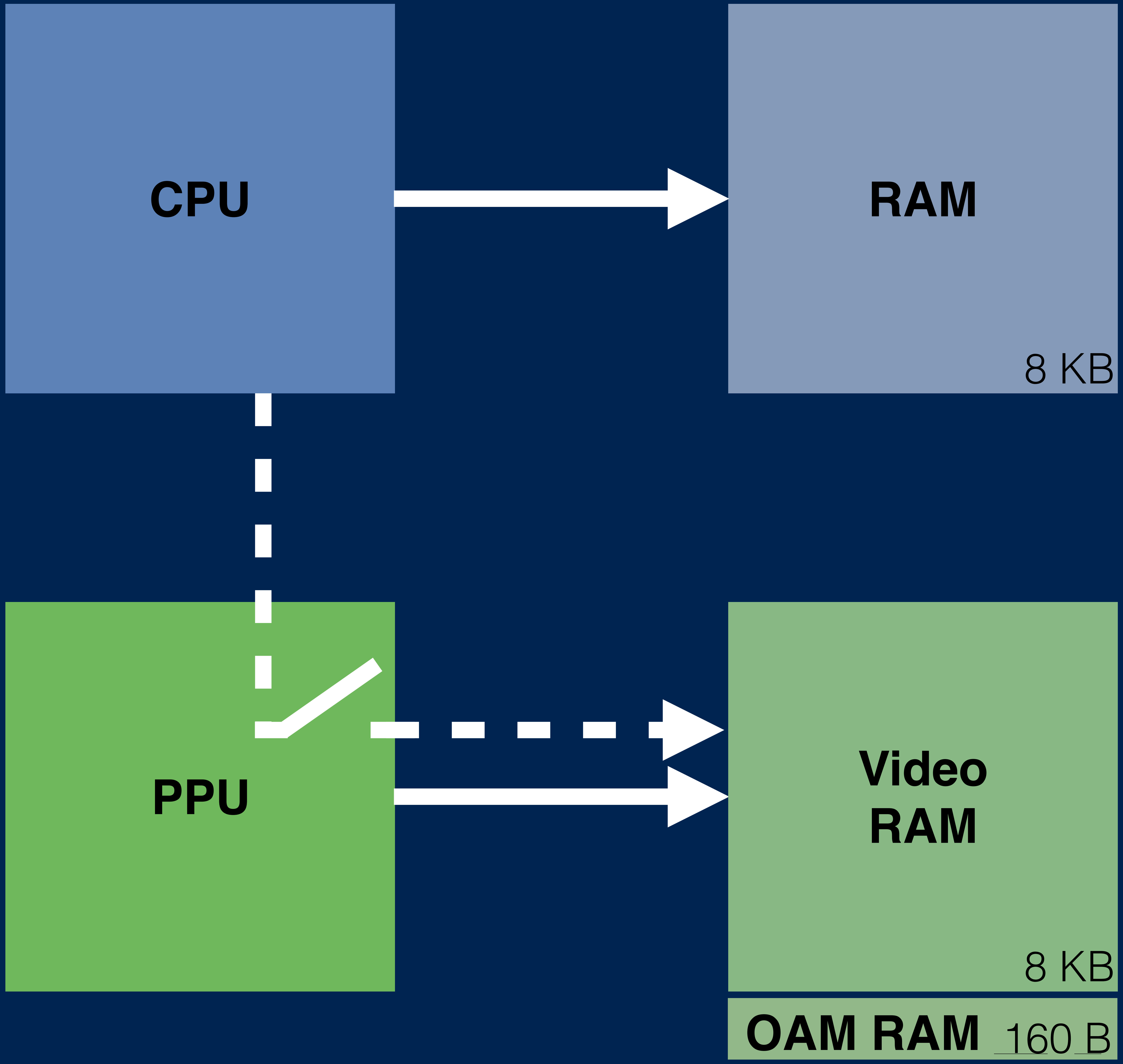
VRAM Access



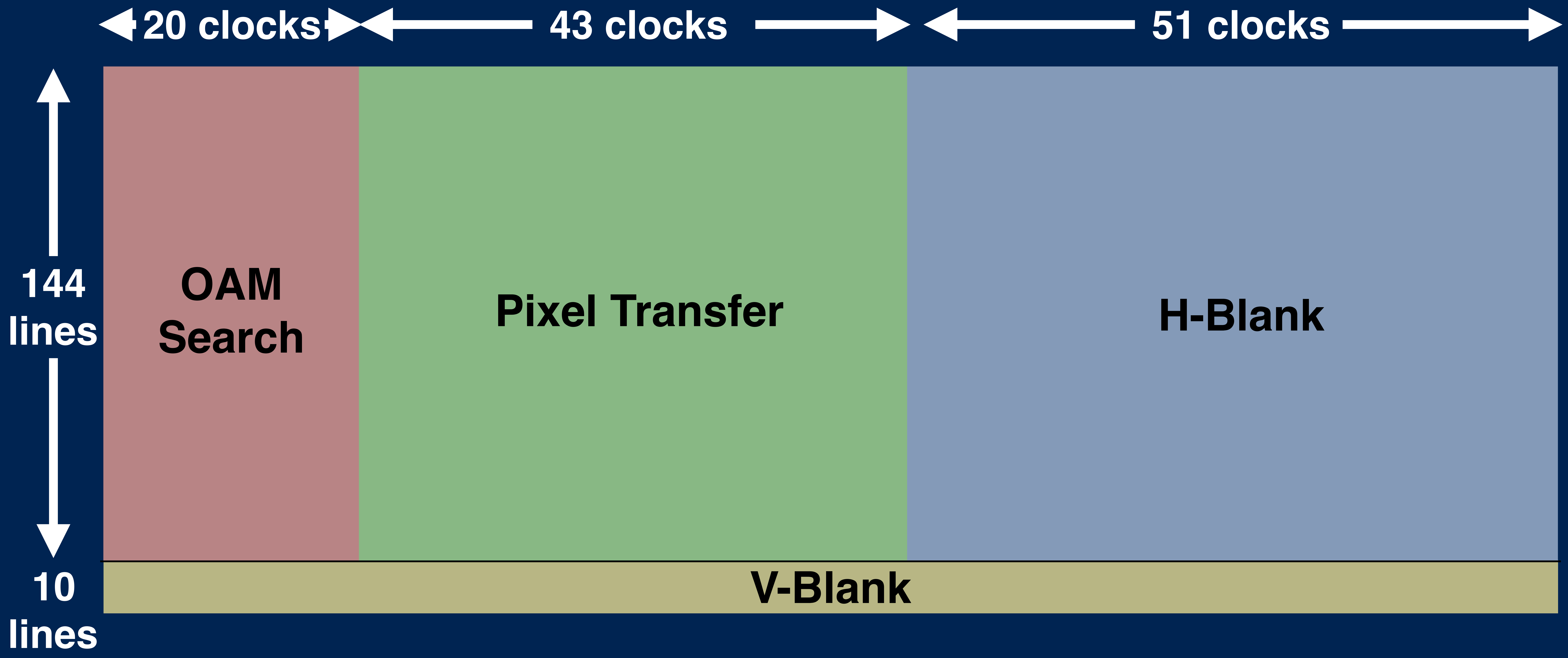
VRAM Access



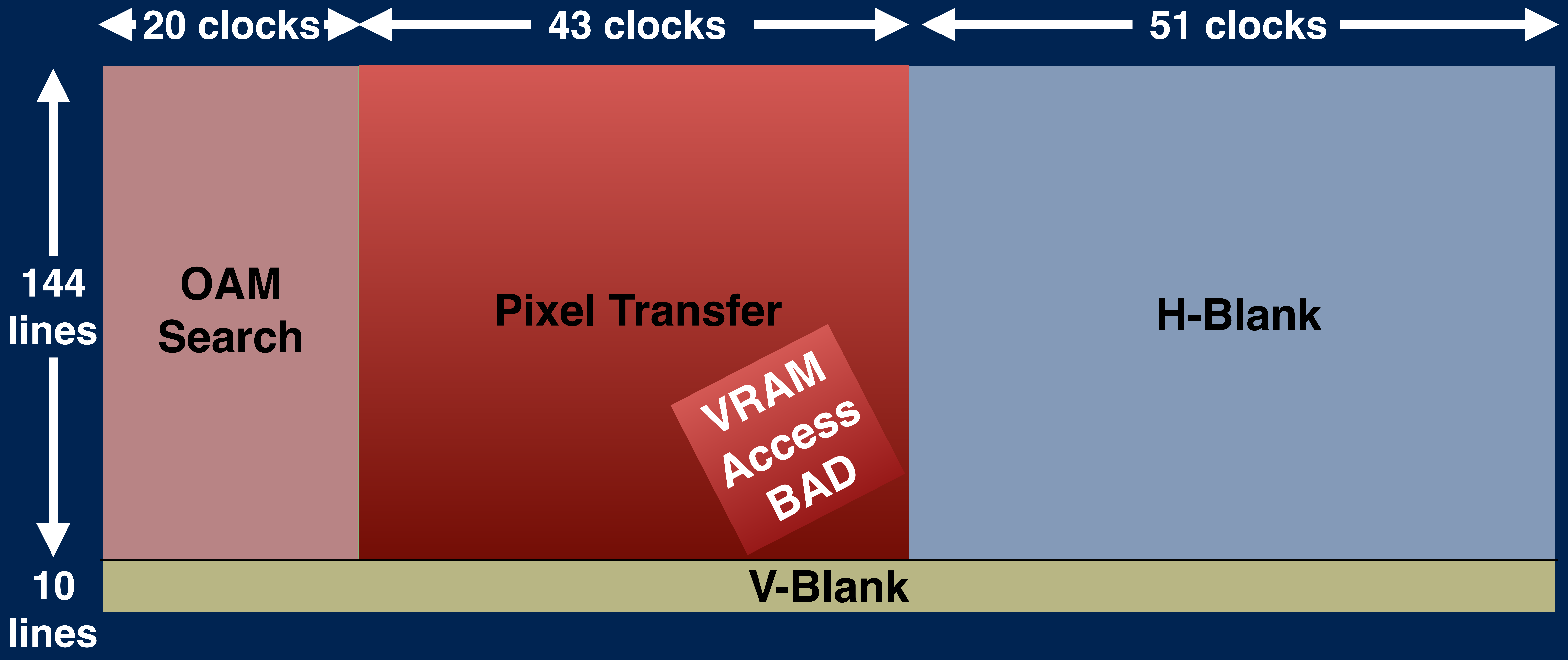
VRAM Access



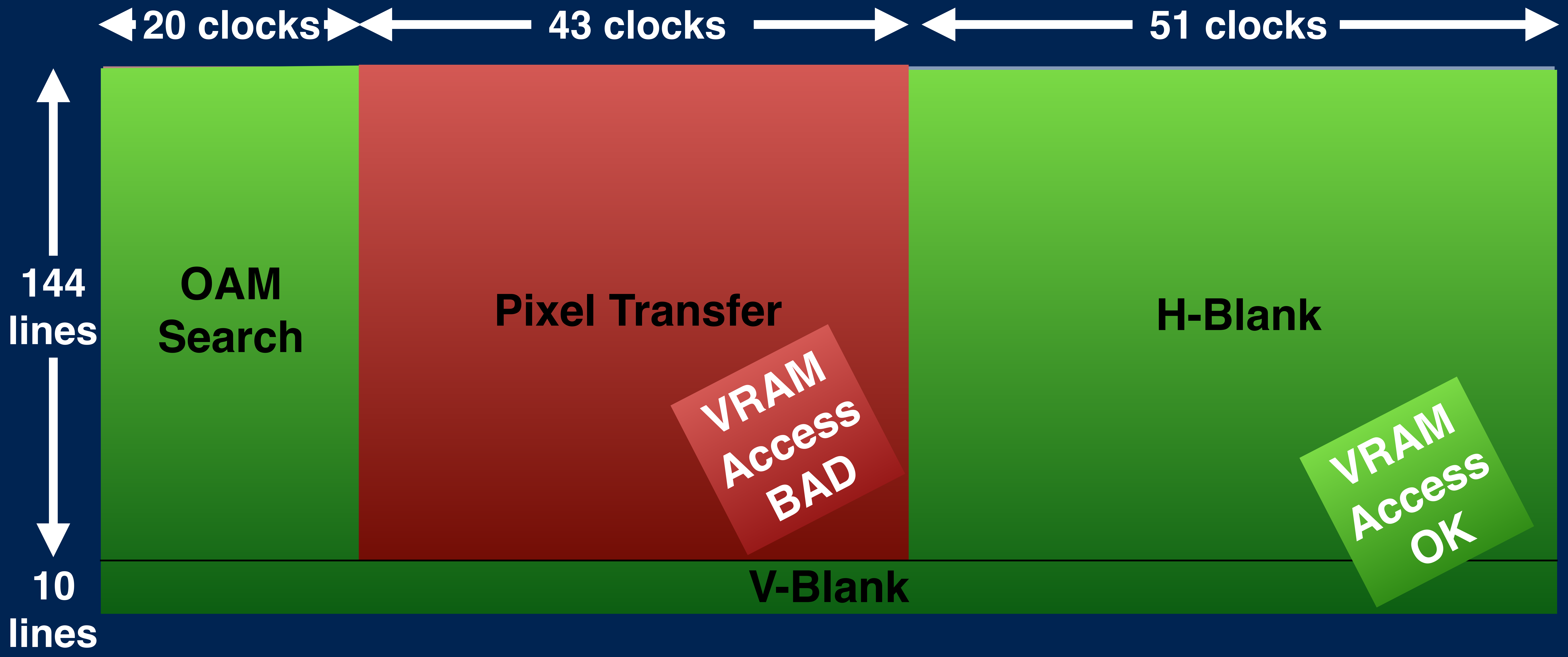
VRAM Access

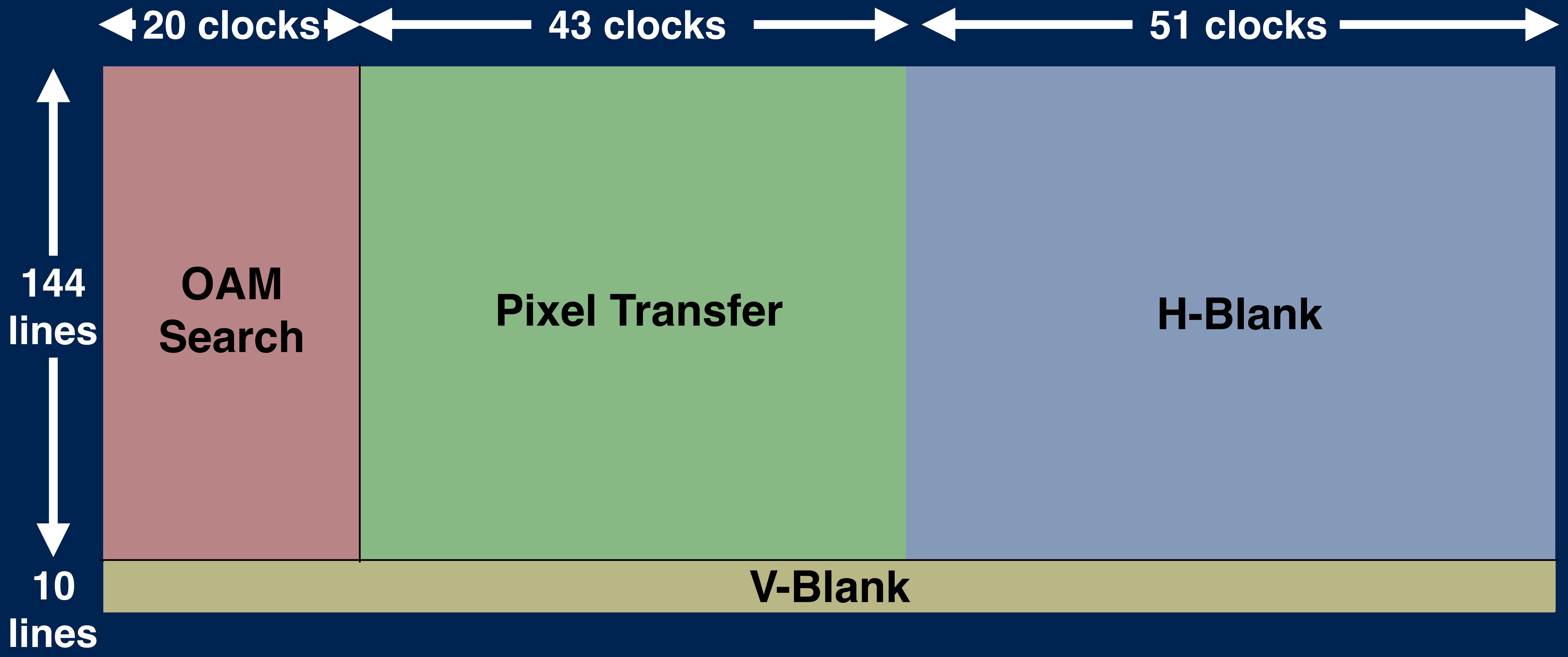

20 clocks
43 clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
10 lines
V-Blank

VRAM Access

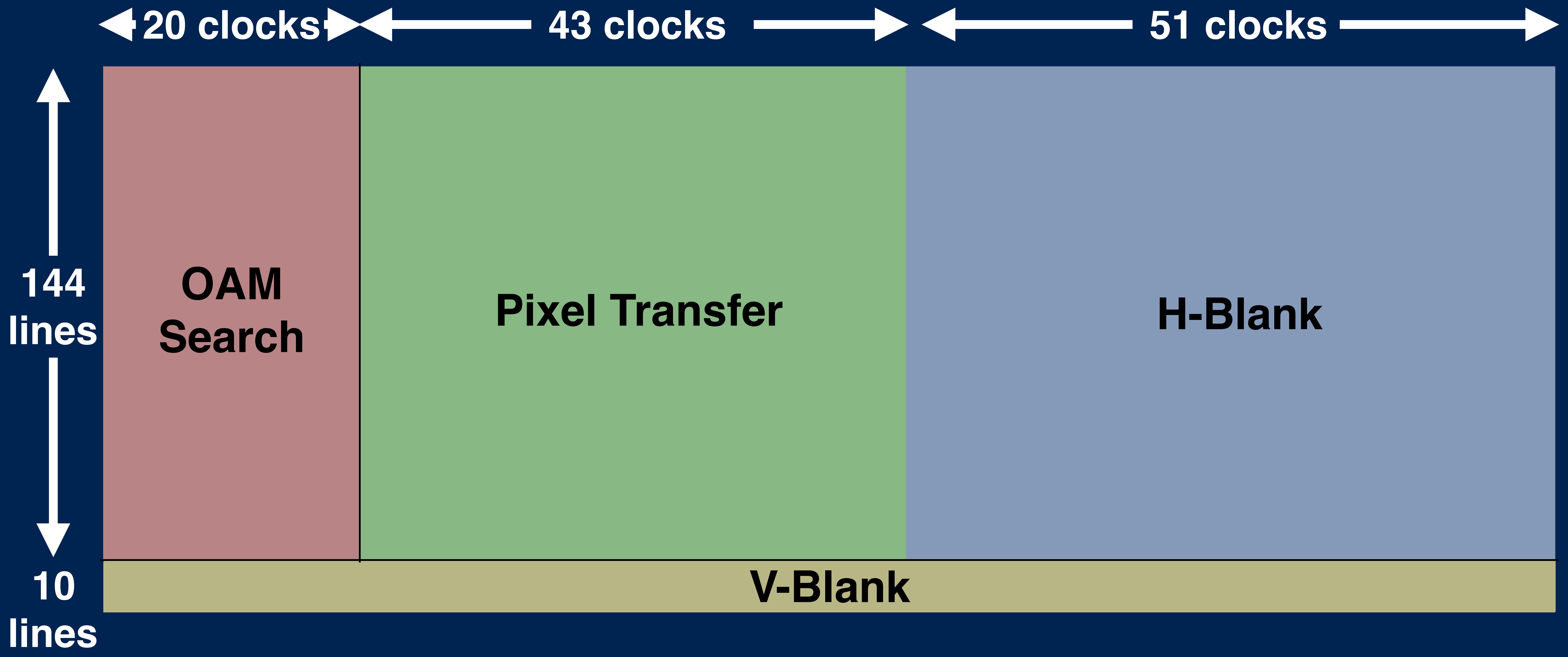

20 clocks
43 clocks
51 clocks
144 lines
10 lines
OAM Search
Pixel Transfer
H-Blank
VRAM Access BAD
V-Blank

VRAM Access


20 clocks
43 clocks
51 clocks
144 lines
10 lines
OAM Search
Pixel Transfer
H-Blank
V-Blank
VRAM Access BAD
VRAM Access OK

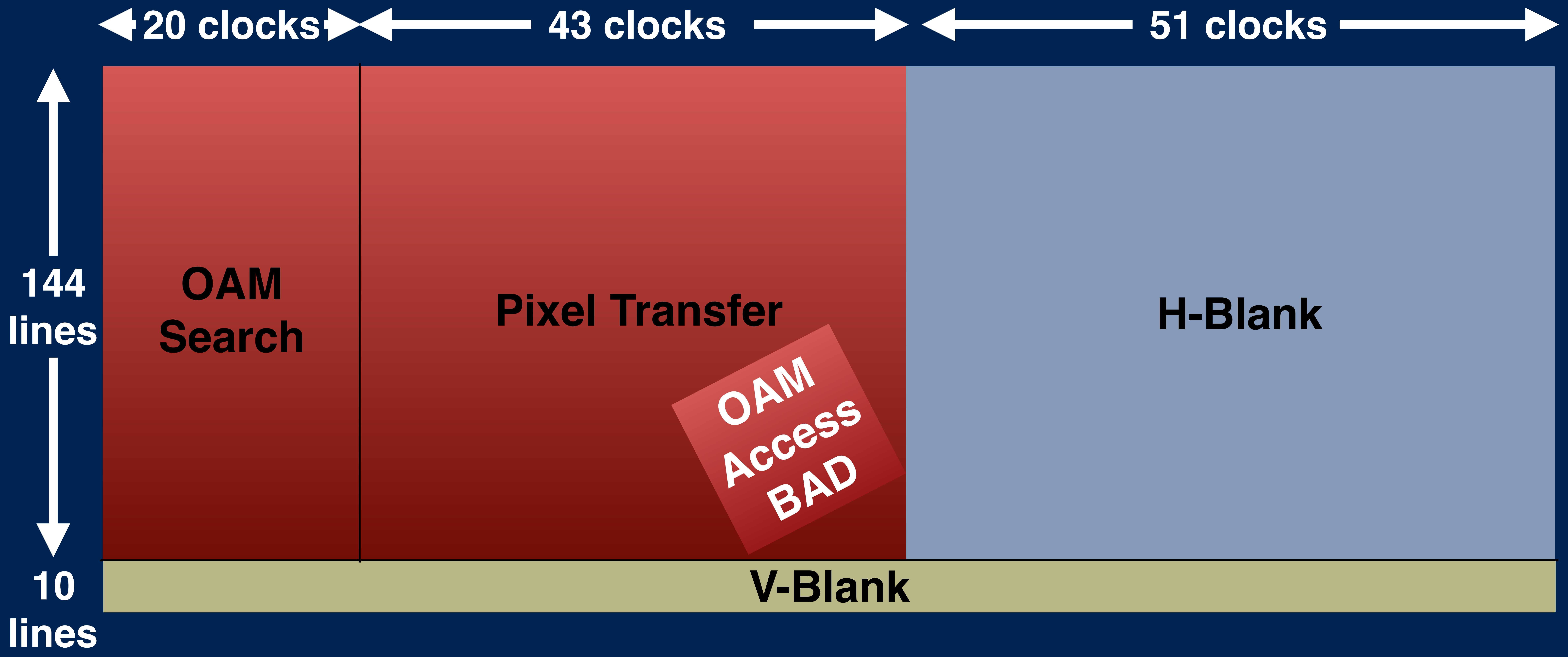

20 clocks
43 clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
10 lines
V-Blank

OAM Access

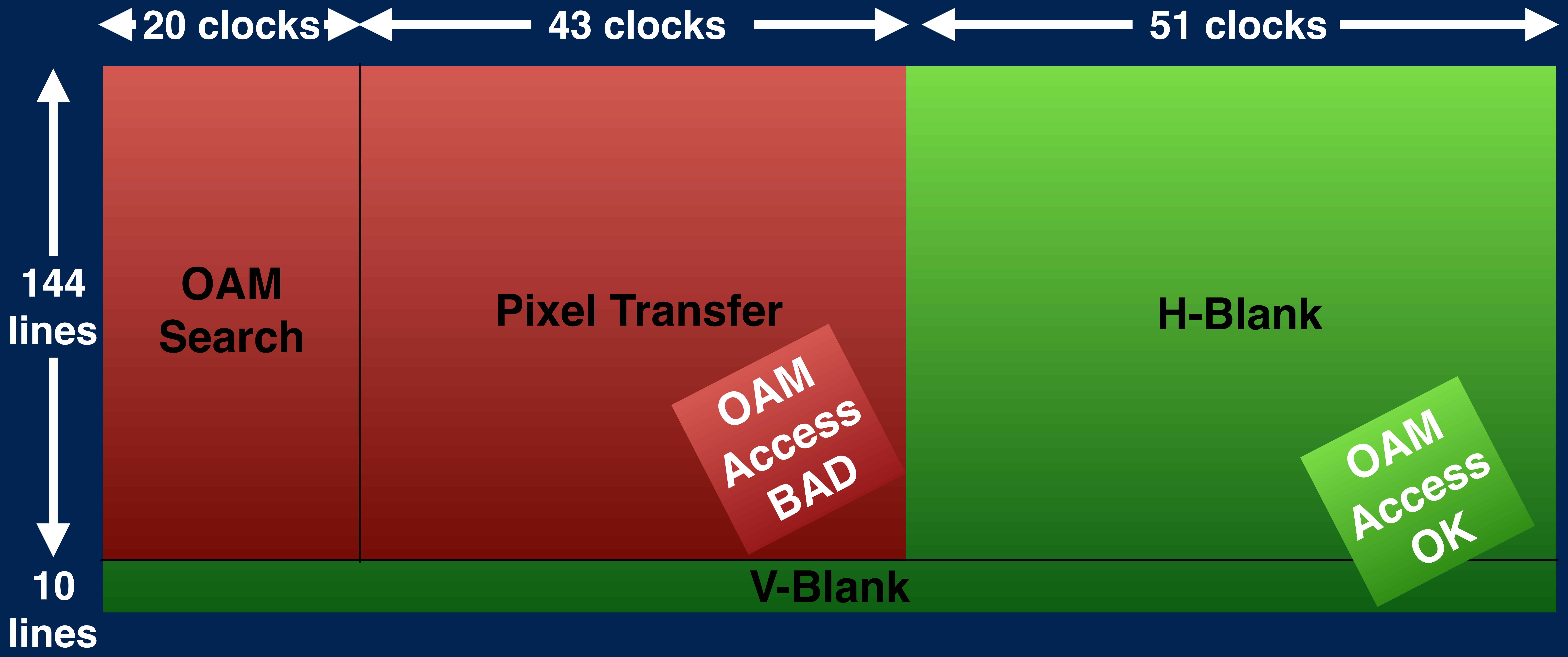


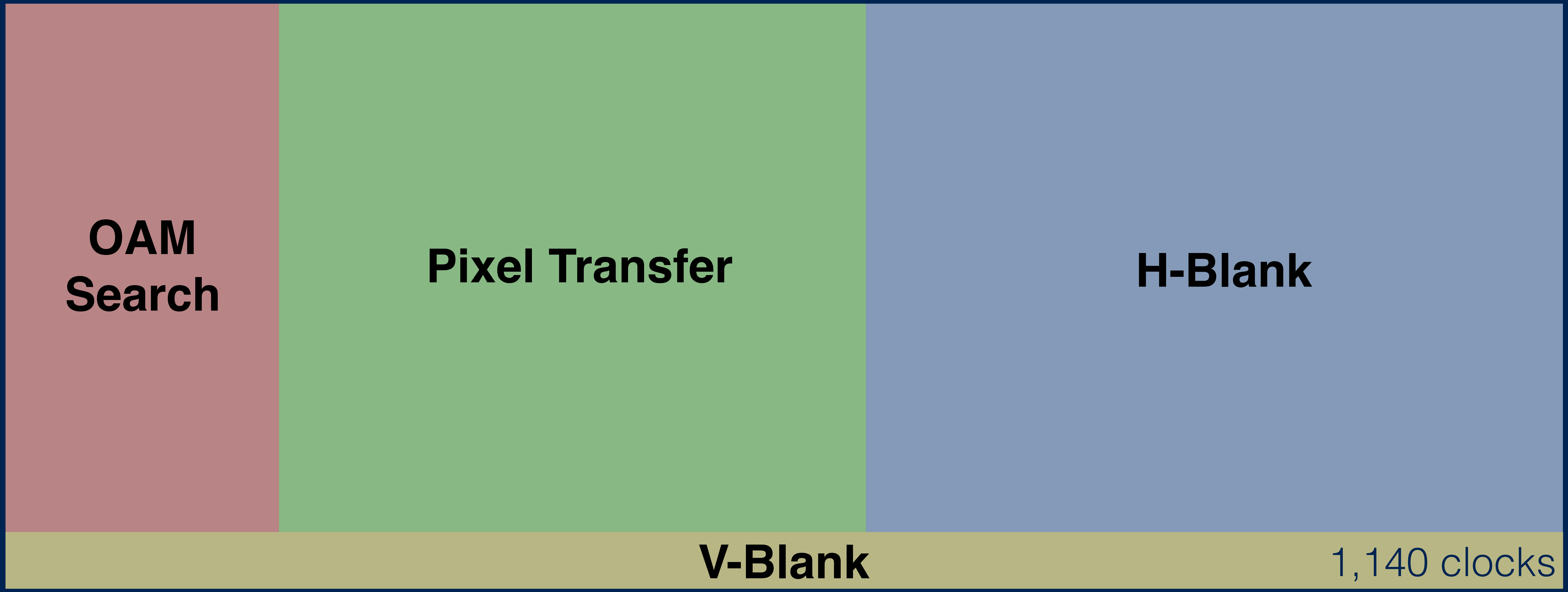
20 clocks
43 clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
10 lines
V-Blank

OAM Access

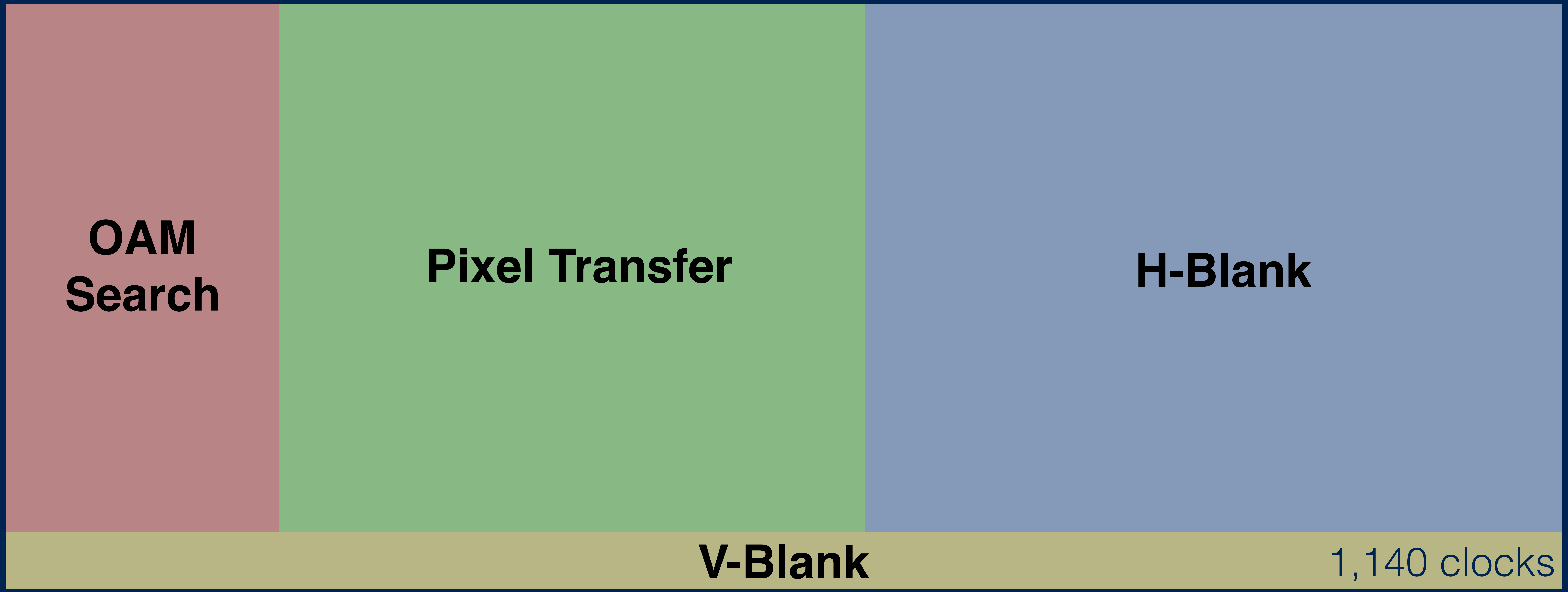

20 clocks
43 clocks
51 clocks
144 lines
10 lines
OAM Search
Pixel Transfer
H-Blank
OAM Access BAD
V-Blank

OAM Access

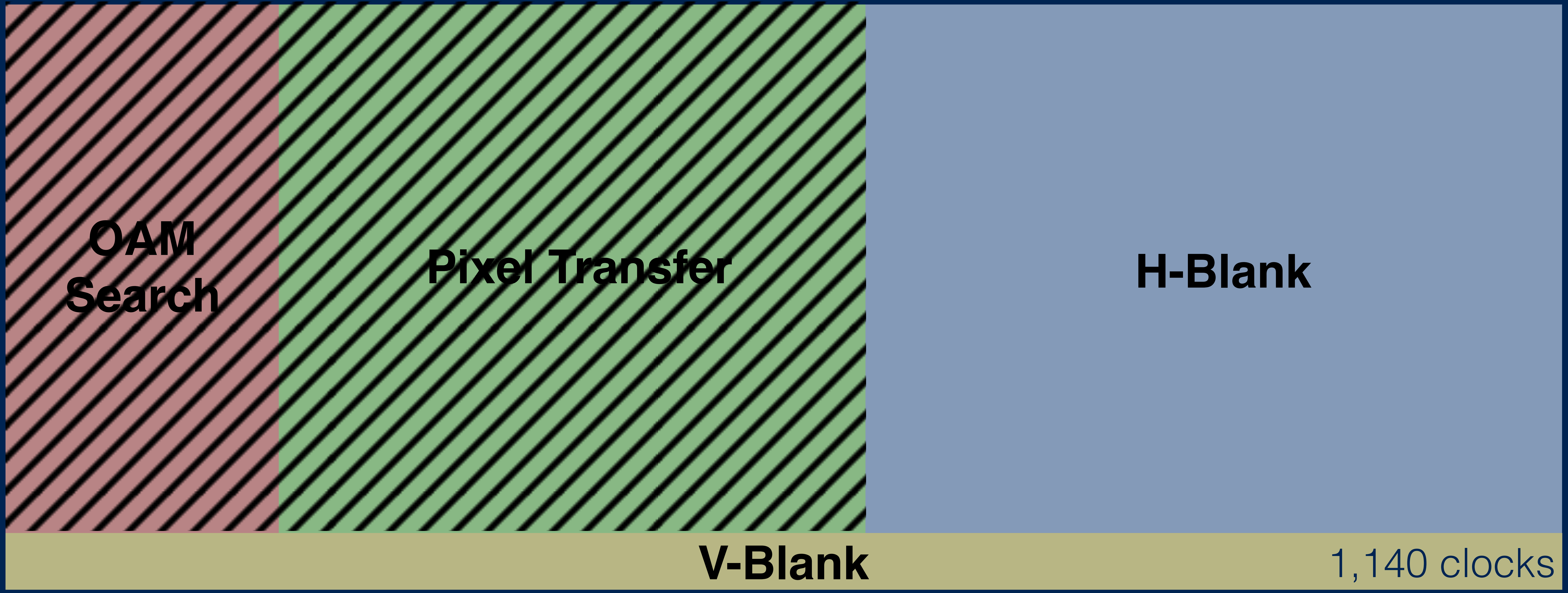

20 clocks
43 clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
OAM Access BAD
OAM Access OK
10 lines
V-Blank


OAM
Search
Pixel Transfer
H-Blank
V-Blank
1,140 clocks

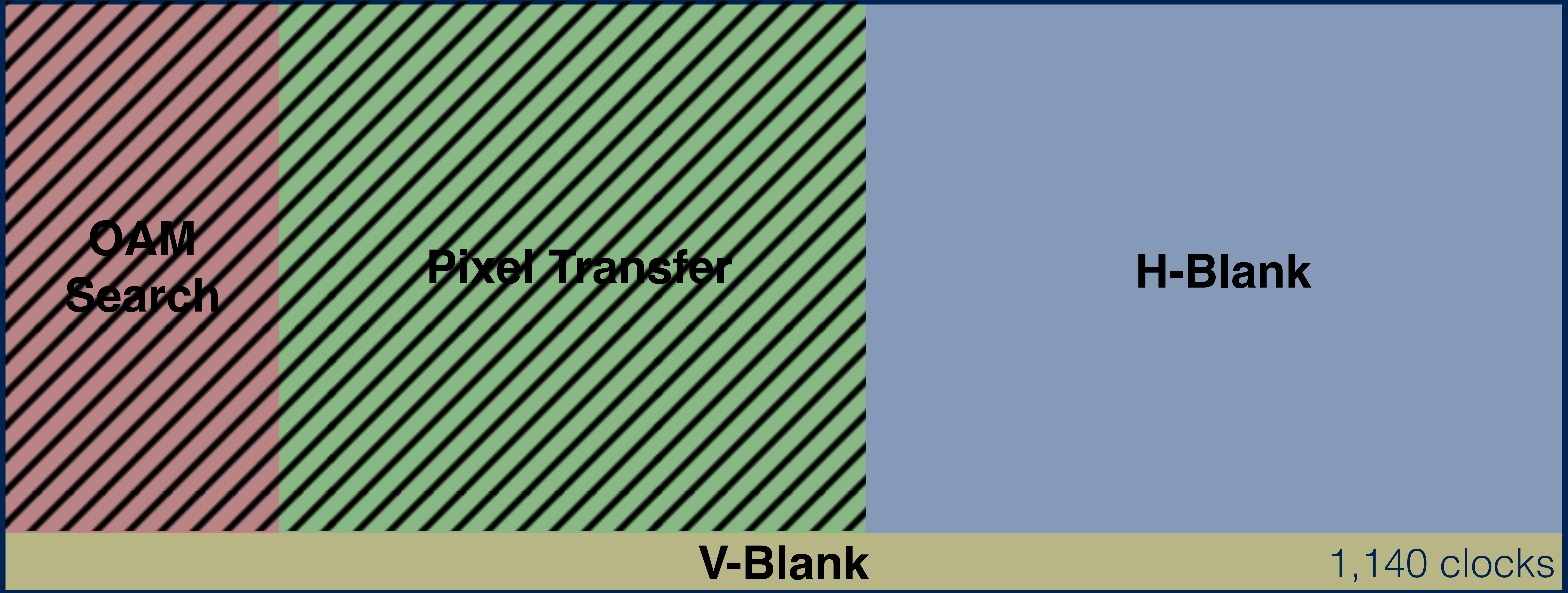
Game Timing



Game Timing

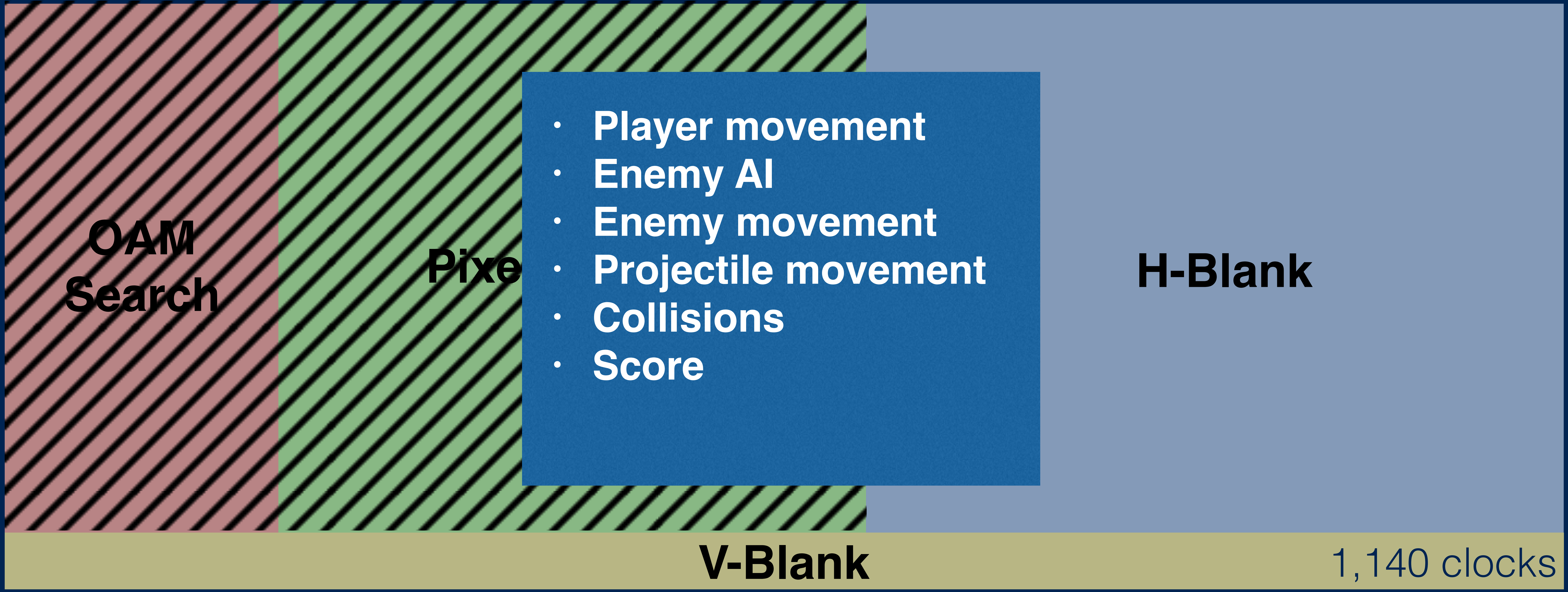


Game Timing



- **Move background map**

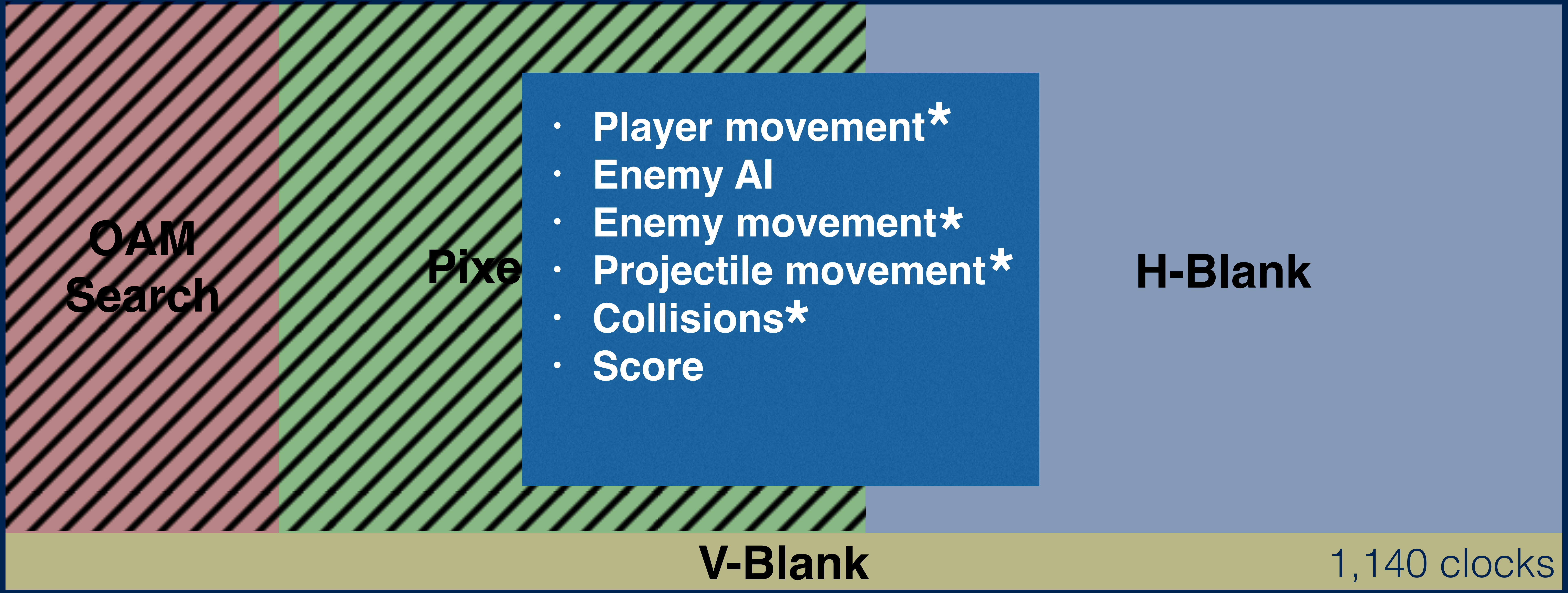
Game Timing



- Player movement
- Enemy AI
- Enemy movement
- Projectile movement
- Collisions
- Score

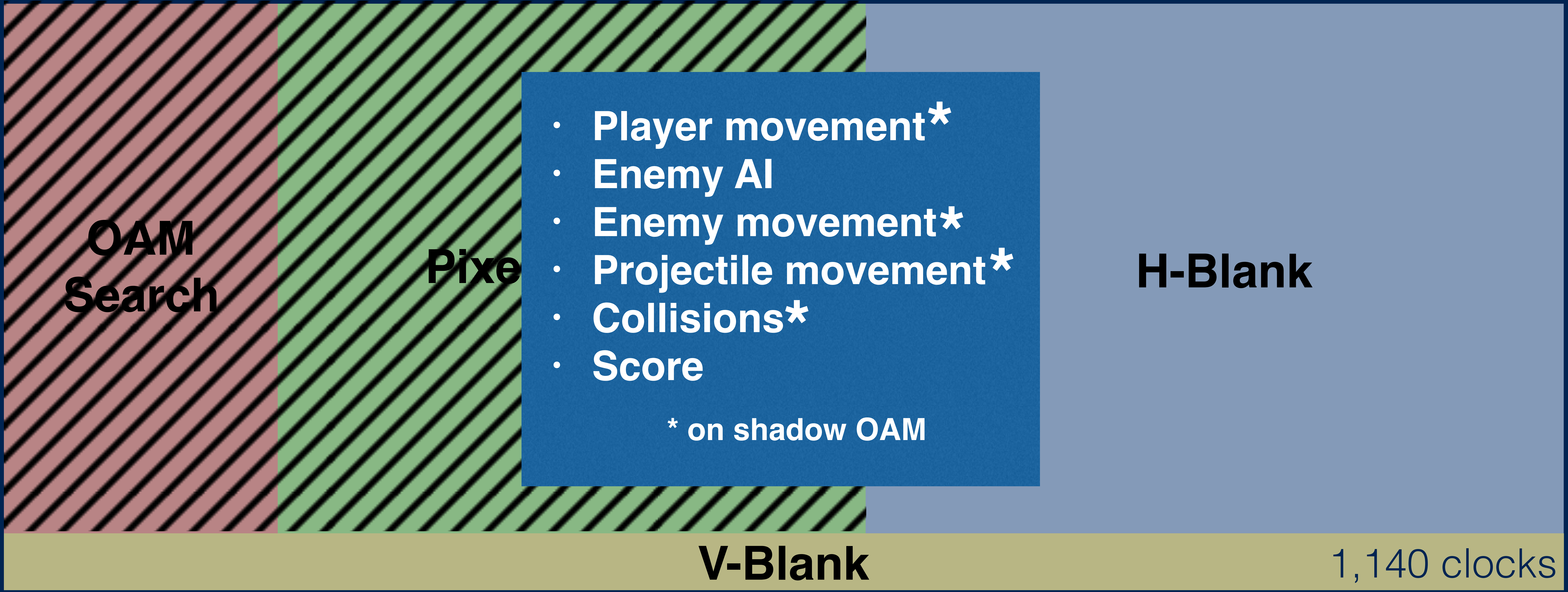
- Move background map

Game Timing



- Move background map

Game Timing

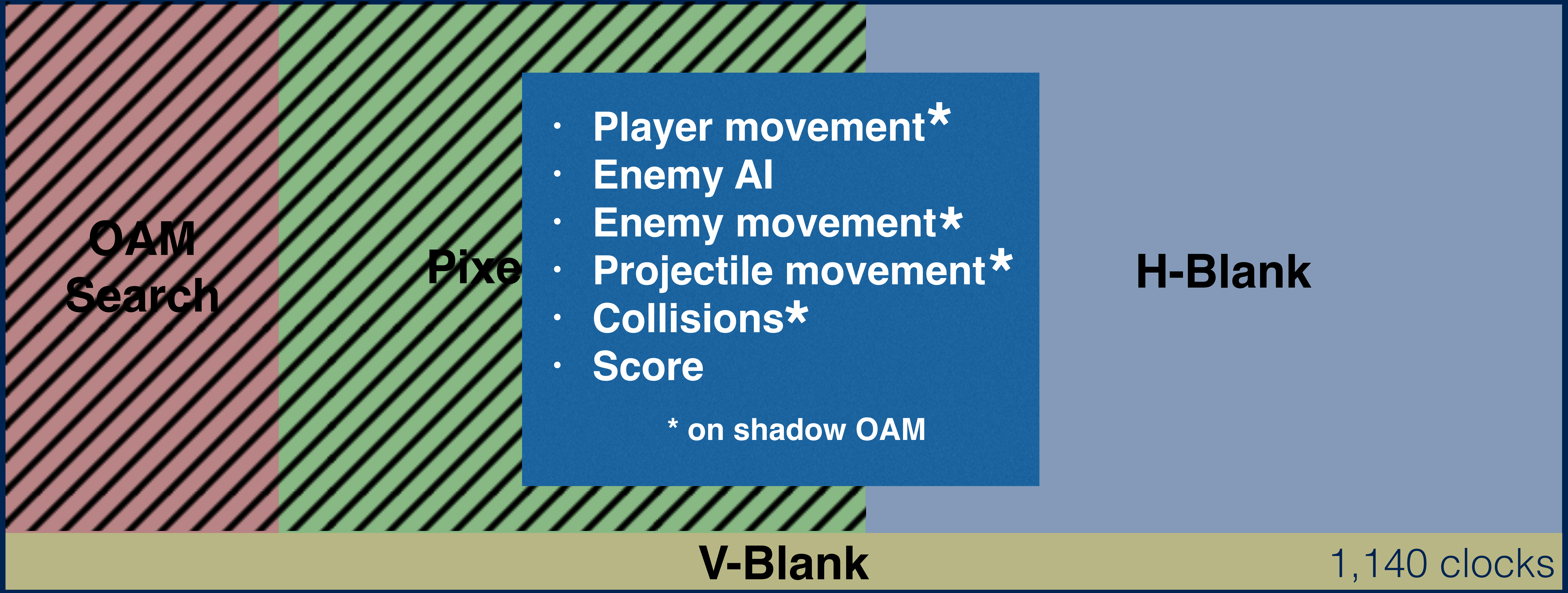


- Player movement*
- Enemy AI
- Enemy movement*
- Projectile movement*
- Collisions*
- Score

* on shadow OAM

- Move background map

Game Timing

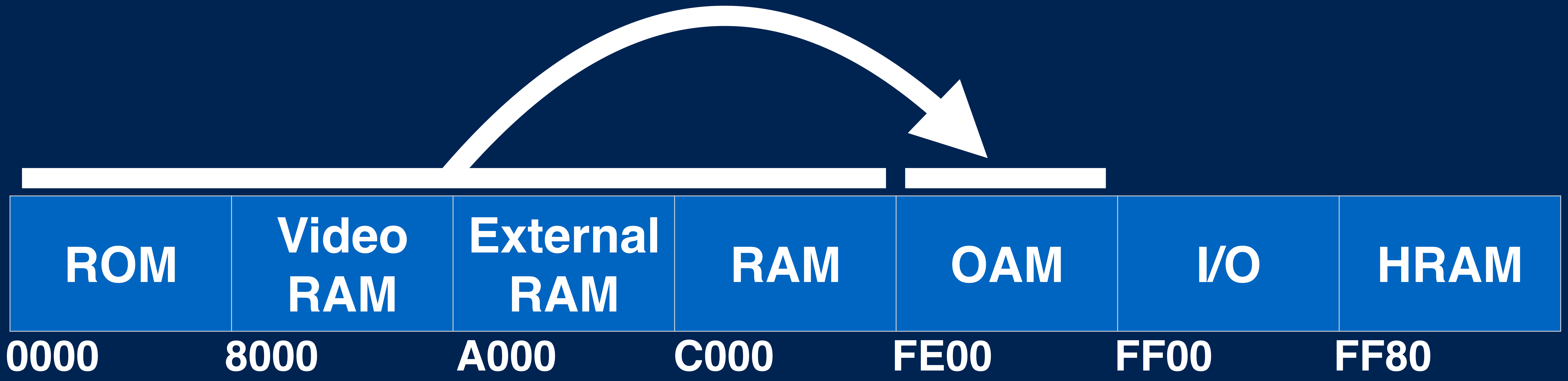


- Player movement*
- Enemy AI
- Enemy movement*
- Projectile movement*
- Collisions*
- Score

* on shadow OAM

- Move background map
- Copy shadow OAM to OAM

ROM	Video RAM	External RAM	RAM	OAM	I/O	HRAM
0000	8000	A000	C000	FE00	FF00	FF80


ROM
Video RAM
External RAM
RAM
OAM
I/O
HRAM
0000
8000
A000
C000
FE00
FF00
FF80

OAM DMA

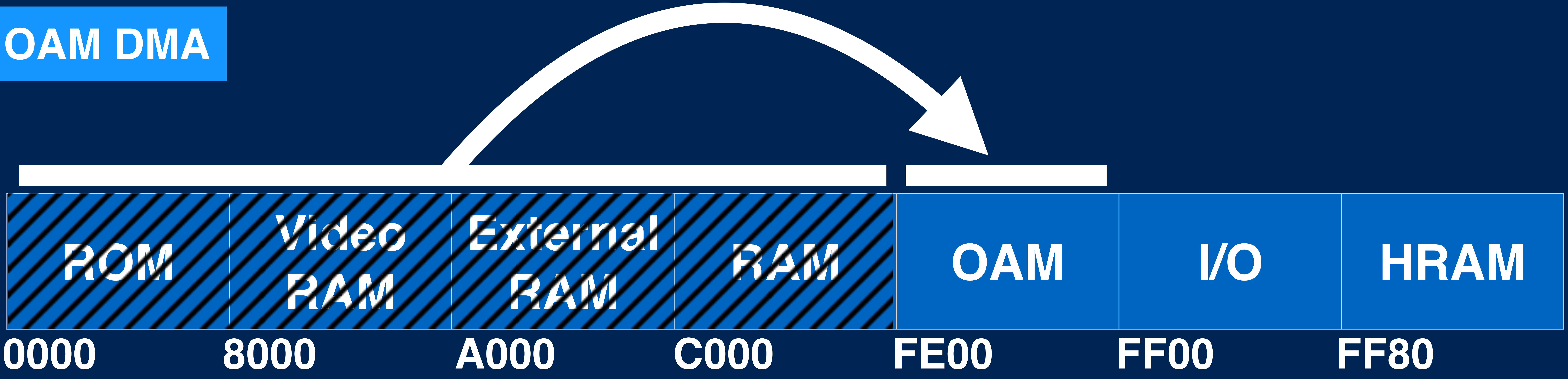
ROM	Video RAM	External RAM	RAM	OAM	I/O	HRAM
0000	8000	A000	C000	FE00	FF00	FF80

```
ld ($FF46), a
```

160 clocks

FF46	DMA	DMA Transfer and Start

OAM DMA

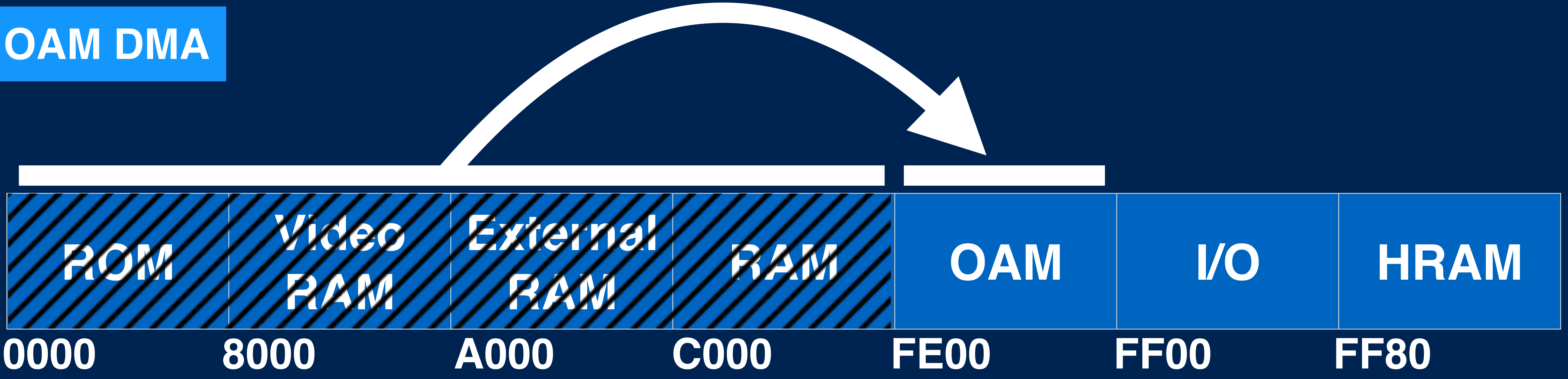



```
ld ($FF46), a
```

160 clocks

FF46	DMA	DMA Transfer and Start

OAM DMA

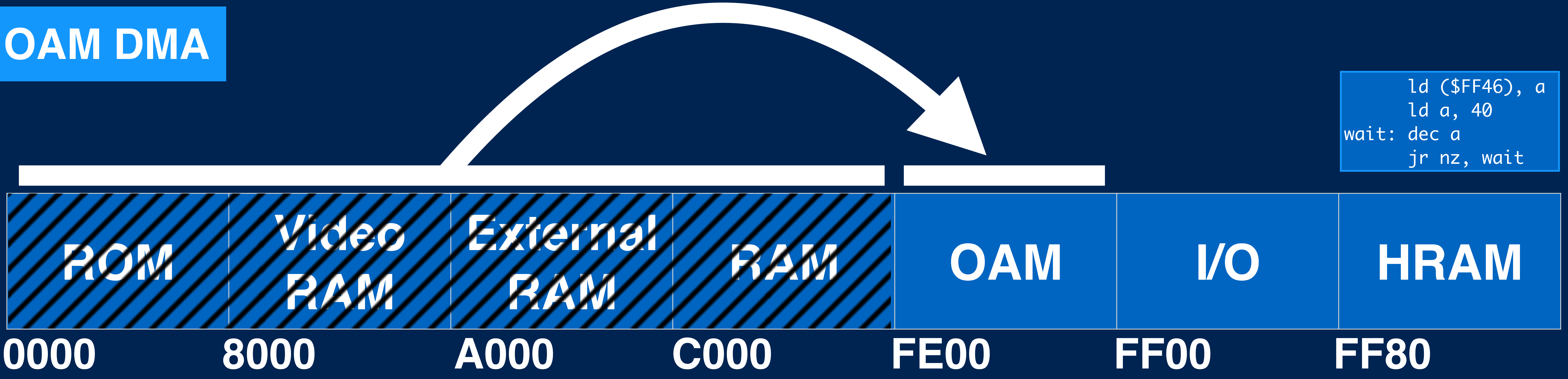



```
       ld ($FF46), a
       ld a, 40
wait:  dec a
       jr nz, wait
```

160 clocks

FF46	DMA	DMA Transfer and Start

OAM DMA




```
      ld ($FF46), a
      ld a, 40
wait: dec a
      jr nz, wait
```

FF46	DMA	DMA Transfer and Start

Pixel Processing Unit

Pixel Processing Unit

Pixel Pipeline

Fine Print

A lot of this information is based on experiments with a custom Game Boy emulator. The results match Game Boy timing traces but could not be verified yet. Also, some information on the following slides is slightly simplified.


Pixel FIFO
LCD

Pixel FIFO
LCD


Pixel FIFO
LCD

Pixel FIFO
LCD


Pixel FIFO
LCD


Pixel FIFO
LCD


Pixel FIFO
LCD


Pixel FIFO
LCD


Pixel FIFO
LCD


Pixel FIFO
LCD


Pixel FIFO
LCD


Pixel FIFO
LCD


Fetch
9802
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


Fetch
9802
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


Fetch
9802
Read Tile #
Read Data 0
Read Data 1
Background Map
9802: 04 EA 2E
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
Read Data 1
Background Map
9802: 04 EA 2E
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
Read Data 1
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
Read Data 1
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
Read Data 1
Tile Data
8040: 02 FE 6A
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
02
Read Data 1
Tile Data
8040: 02 FE 6A
Pixel FIFO
LCD

Fetch
9802
Read Tile #
04
Read Data 0
02
Read Data 1
Tile Data
8040: 02 FE 6A
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
02
Read Data 1
Tile Data
8040: 02 FE 6A
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
02
Read Data 1
FE
Tile Data
8040: 02 FE 6A
Pixel FIFO
LCD


Fetch
9802
Read Tile #
04
Read Data 0
02
Read Data 1
FE
Pixel FIFO
LCD


Fetch
9802
Read Tile #
Read Data 0
02
Read Data 1
FE
Pixel FIFO
LCD


Fetch
9802
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD

FIFO
Fetch
Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Push
Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Push
Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Push
Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Push
Fetch
9803
Read Tile #
60
Read Data 0
Read Data 1
Pixel FIFO
LCD

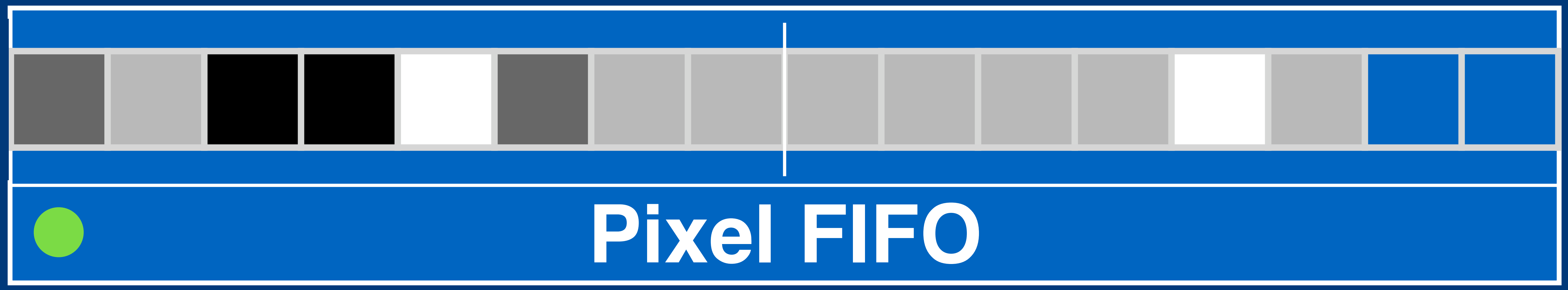
FIFO	Fetch
Push	Read Tile #
Push	

Fetch 9803

Read Tile # 60

→ Read Data 0

Read Data 1



LCD

FIFO	Fetch
Push	Read Tile #
Push	

Fetch 9803

- Read Tile # 60
- → Read Data 0
- Read Data 1

Pixel FIFO

LCD


FIFO
Fetch
Push
Push
Push
Read Tile #
Fetch
9803
Read Tile #
60
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Push
Push
Push
Read Tile #
Fetch
9803
Read Tile #
60
Read Data 0
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Push
Read Tile #
Push
Push
Fetch
9803
Read Tile #
60
Read Data 0
Read Data 1
Pixel FIFO
LCD

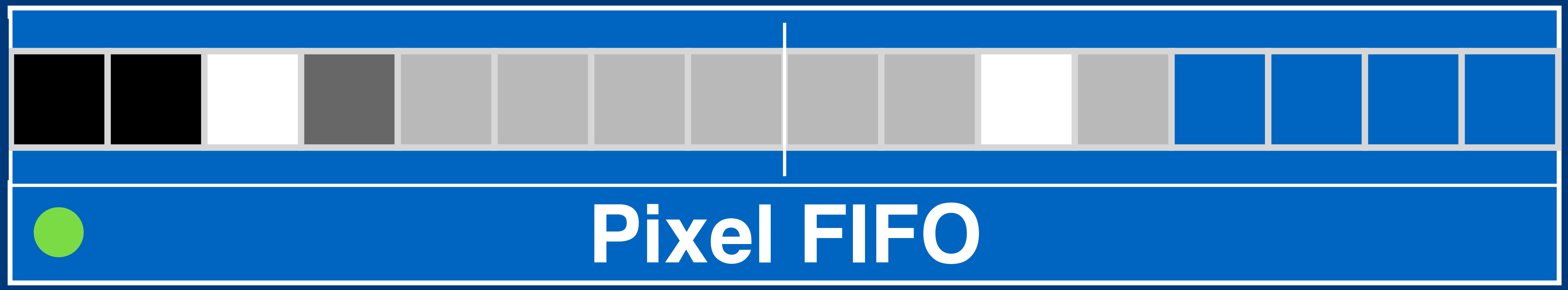
FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	

Fetch 9803

Read Tile # 60

Read Data 0 73

Read Data 1



LCD


FIFO
Fetch
Push
Push
Push
Push
Read Tile #
Read Data 0
Fetch
9803
Read Tile #
60
Read Data 0
73
Read Data 1
Pixel FIFO
LCD

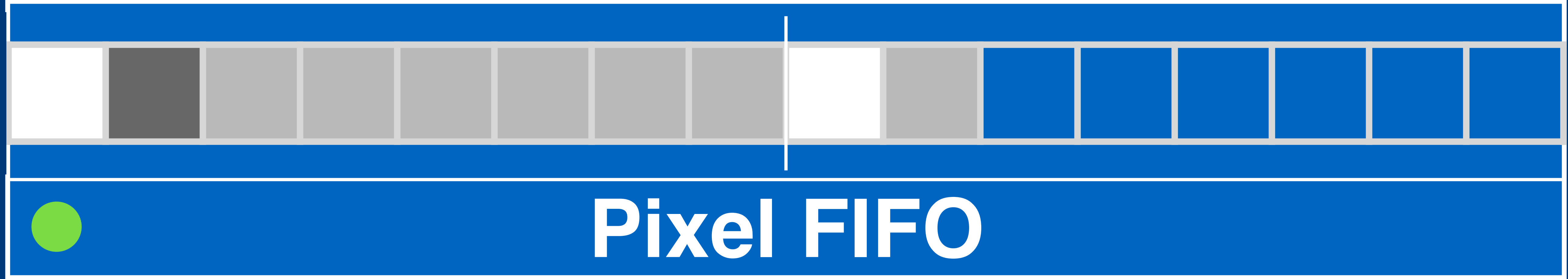

FIFO
Fetch
Push
Push
Push
Push
Push
Read Tile #
Read Data
0
Fetch
9803
Read Tile #
60
Read Data 0
73
Read Data 1
Pixel FIFO
LCD


FIFO
Fetch
Push
Read Tile #
Push
Push
Read Data 0
Push
Push
Fetch
9803
Read Tile #
60
Read Data 0
73
Read Data 1
Pixel FIFO
LCD

FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	

Fetch 9803

Read Tile # 60
Read Data 0 73
→ Read Data 1

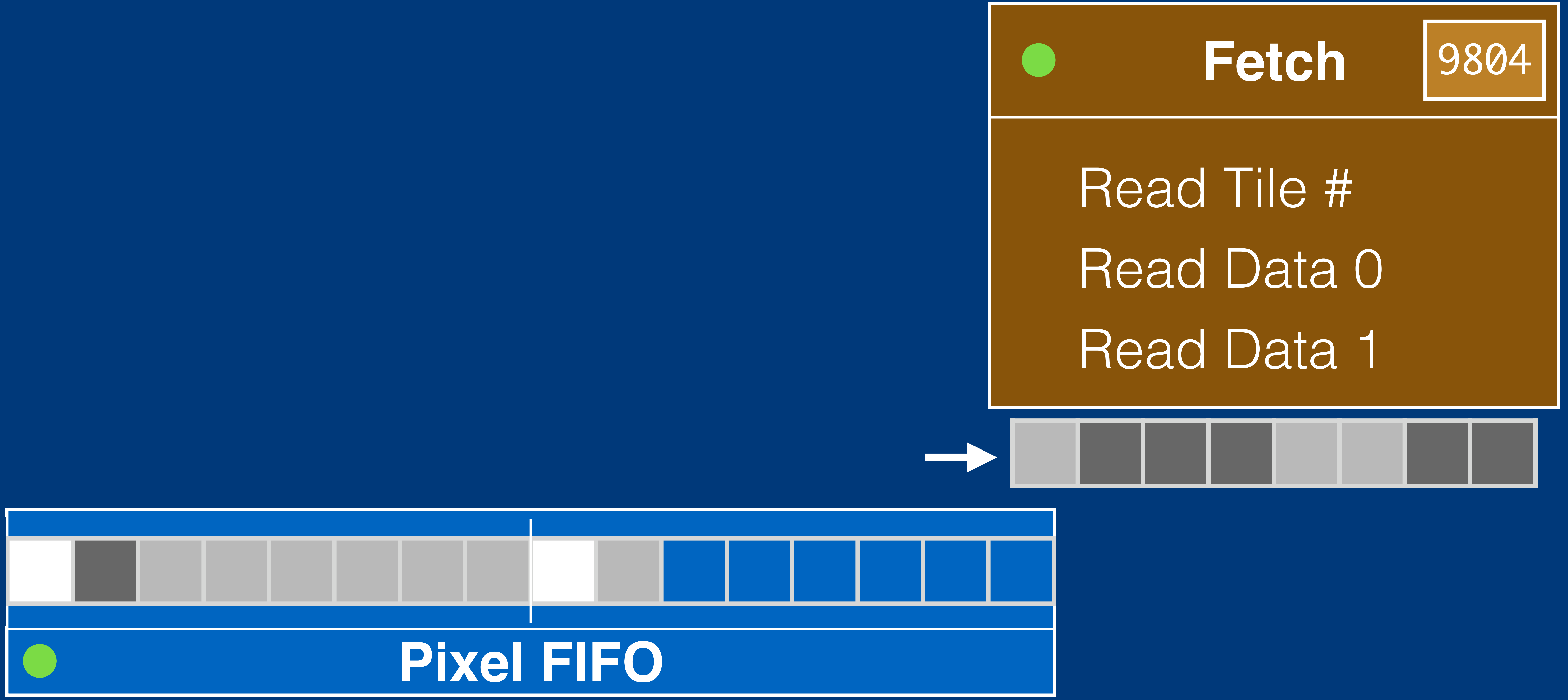


LCD

FIFO
Fetch
Push
Push
Push
Push
Push
Read Tile #
Read Data 0
Fetch
9803
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD

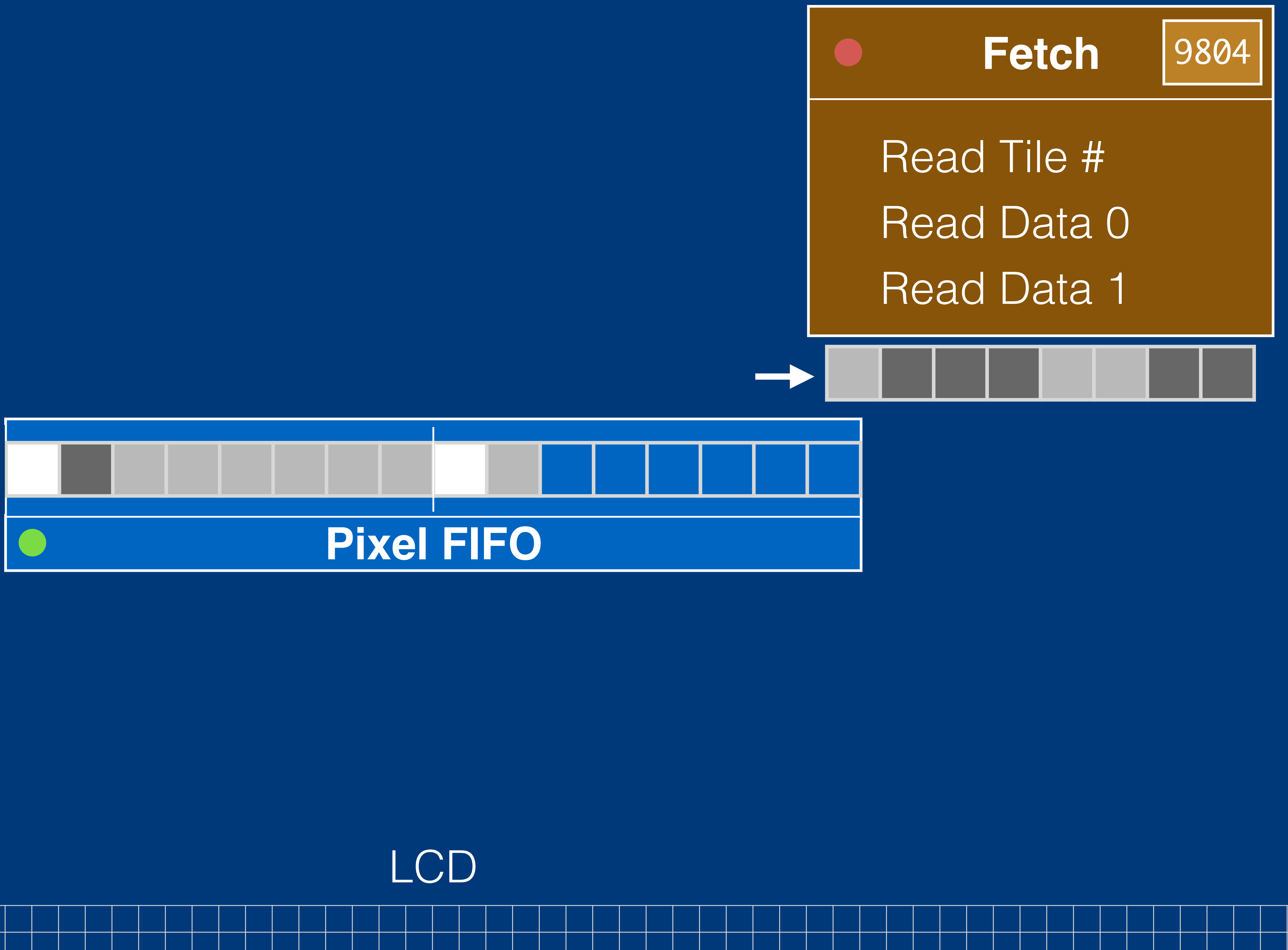

FIFO
Fetch
Push
Read Tile #
Push
Push
Read Data 0
Push
Push
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD

FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	Read Data 1
Push	

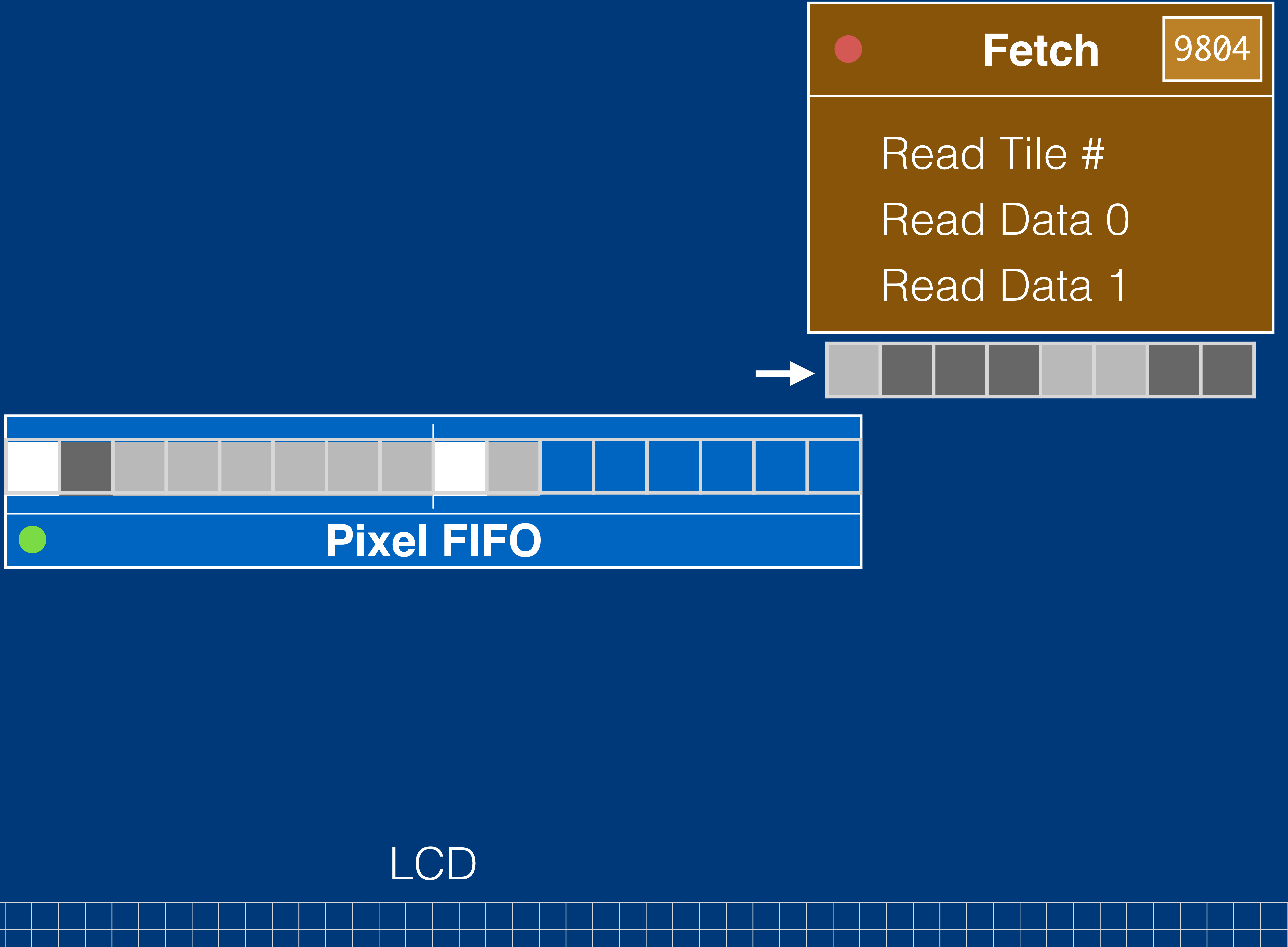


LCD

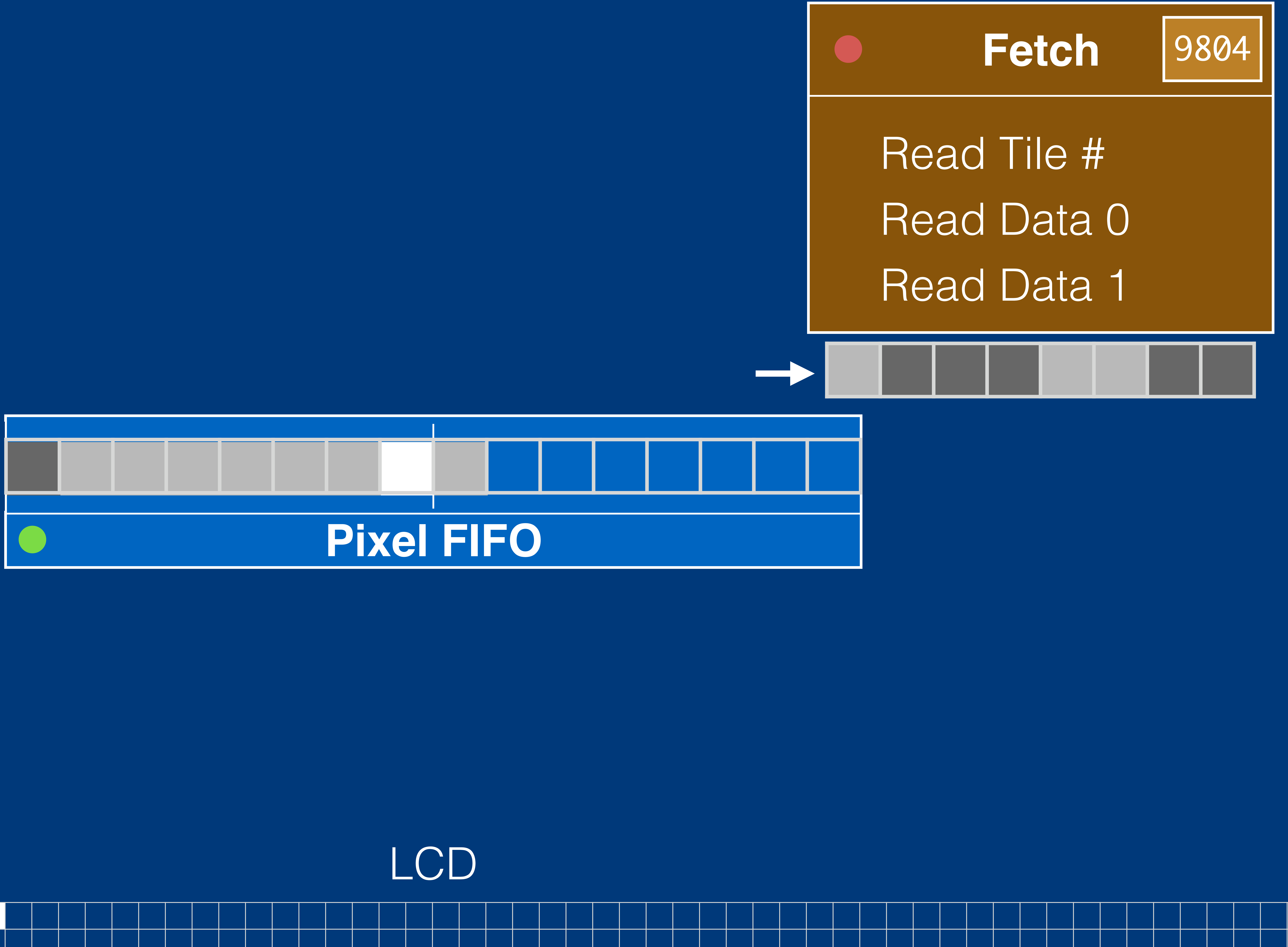
FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	Read Data 1
Push	



FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	Read Data 1
Push	



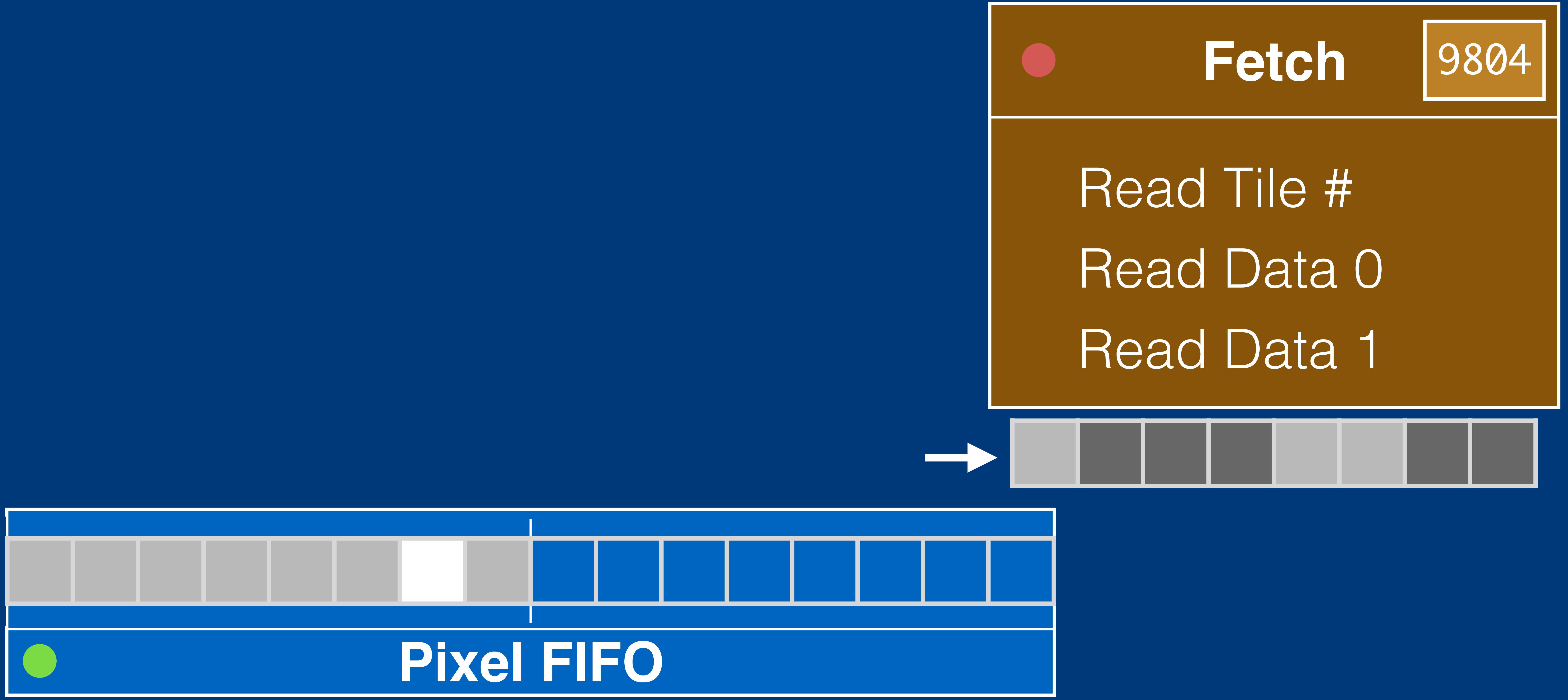
FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	Read Data 1
Push	
Push	




FIFO
Fetch
Push
Read Tile #
Push
Push
Read Data 0
Push
Push
Read Data 1
Push
Push
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD

FIFO
Fetch
Push
Push
Push
Push
Push
Push
Push
Read Tile #
Read Data 0
Read Data 1
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD

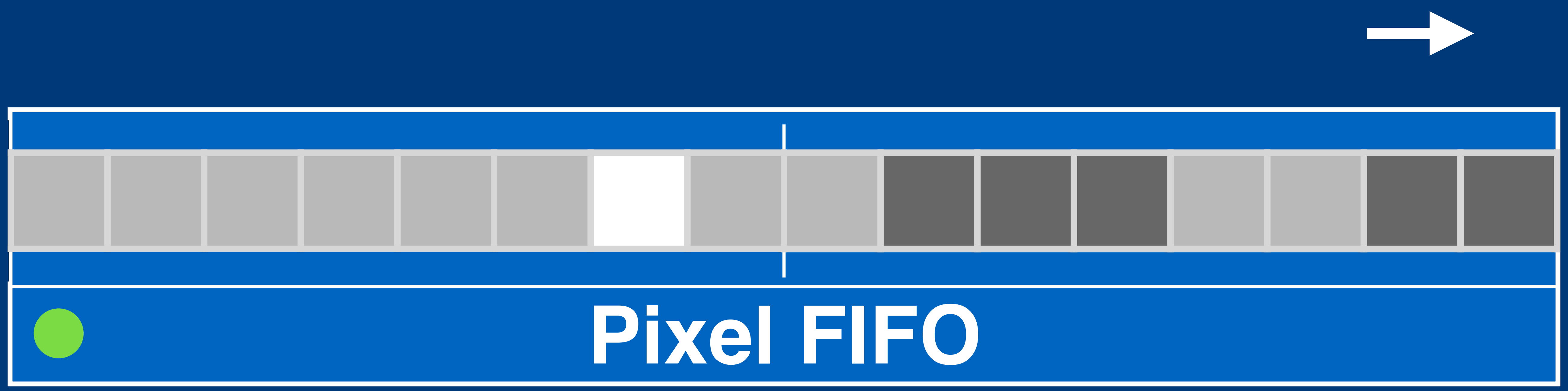
FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	Read Data 1
Push	
Push	Idle
Push	



FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	Read Data 1
Push	
Push	Idle
Push	

Fetch 9804

Read Tile #
Read Data 0
Read Data 1

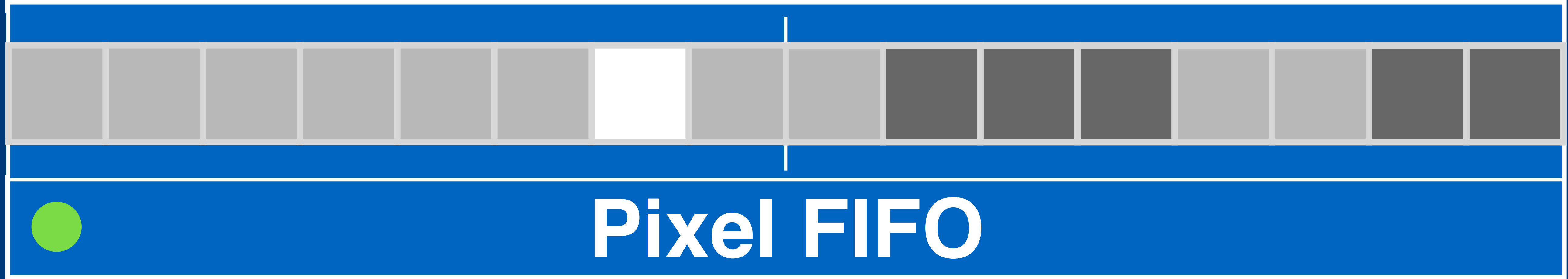


LCD

FIFO	Fetch
Push	Read Tile #
Push	
Push	Read Data 0
Push	
Push	Read Data 1
Push	
Push	Idle
Push	

Fetch 9804

→ Read Tile #
Read Data 0
Read Data 1



LCD


FIFO
Fetch
Push
Push
Push
Push
Push
Push
Push
Push
Read Tile #
Read Data 0
Read Data 1
Idle
4 MHz
2 MHz
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD

FIFO — 4 MHz

- pushes one pixel per clock
- pauses unless it contains more than 8 pixels

Fetch — 2 MHz

- 3 clocks to fetch 8 pixels
- pauses in 4th clock unless space in FIFO

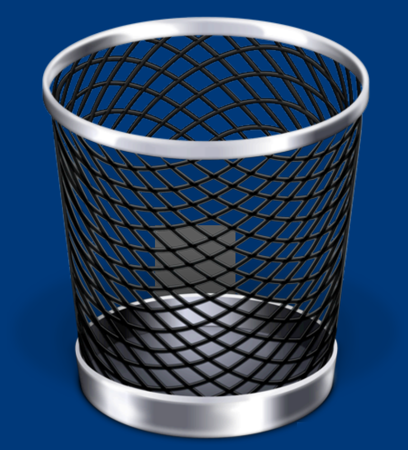

Pixel FIFO
LCD

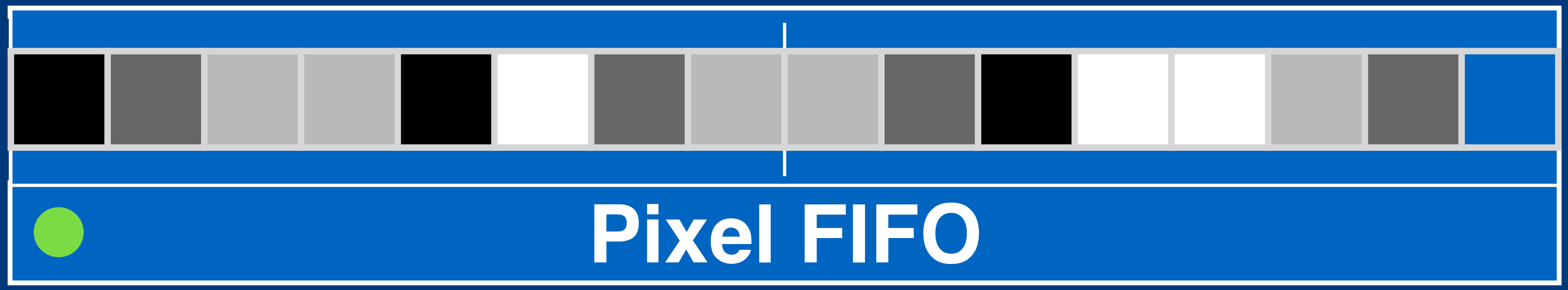
Horizontal Scrolling

SCX = 3

Horizontal Scrolling

SCX = 3





LCD

Horizontal Scrolling

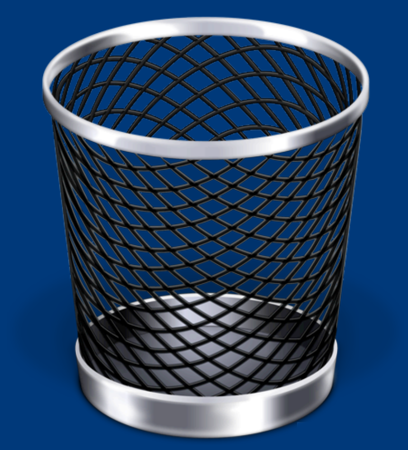
SCX = 3

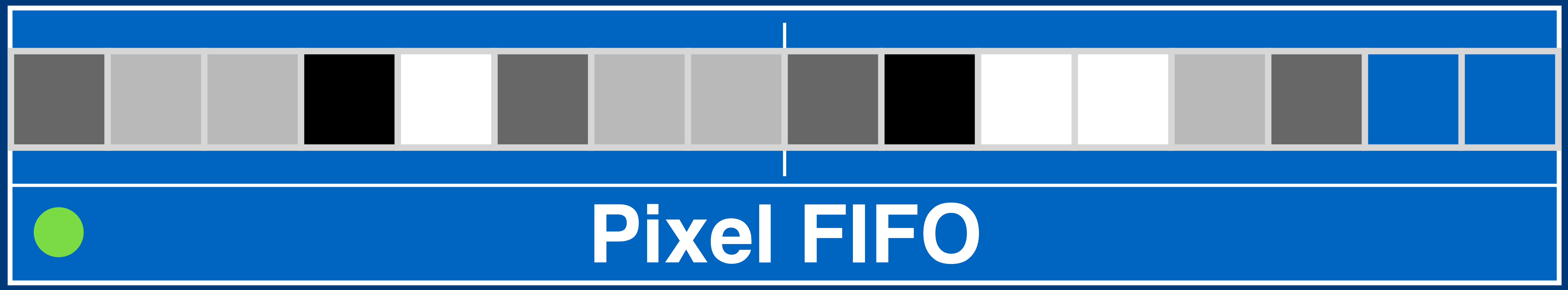
Horizontal Scrolling

SCX = 3

Horizontal Scrolling

SCX = 3

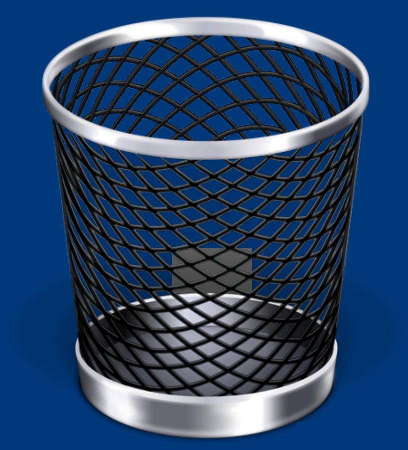


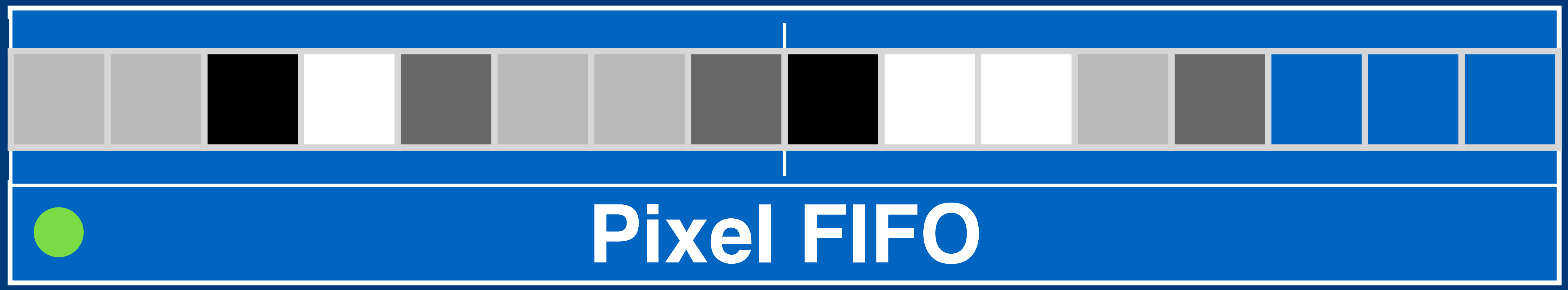


LCD

Horizontal Scrolling

SCX = 3

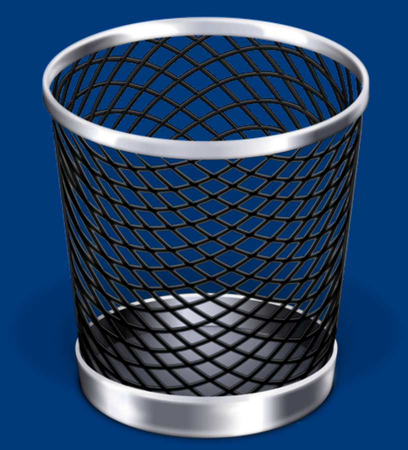


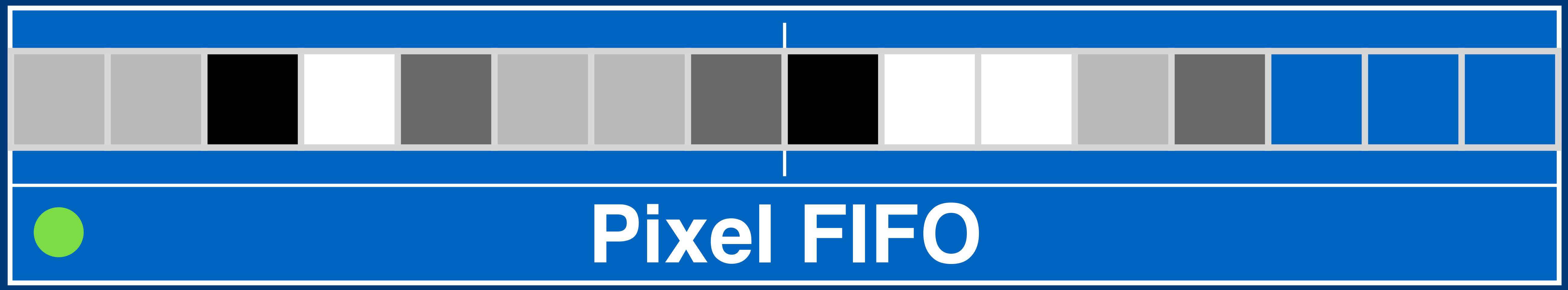


LCD

Horizontal Scrolling

SCX = 3

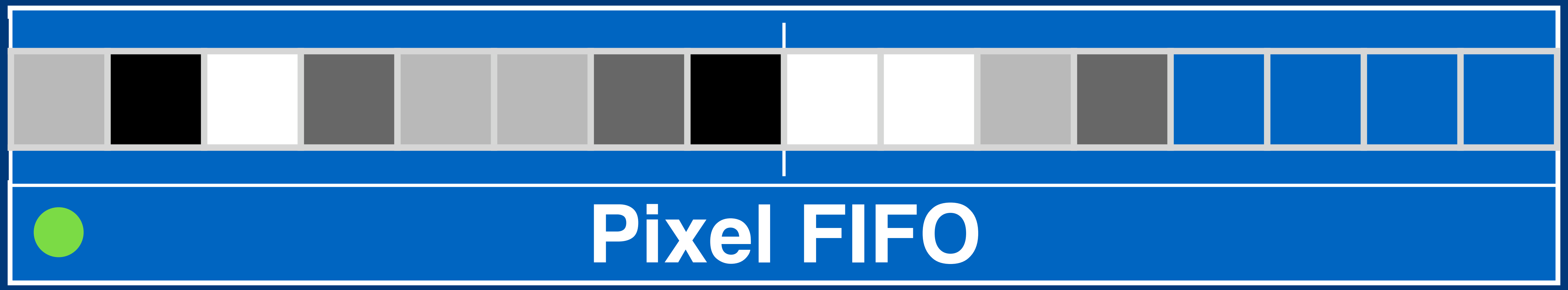




LCD

Horizontal Scrolling

SCX = 3



LCD


Fetch
9816
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


End of Line
Fetch
9816
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


End of Line
Fetch
9816
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
End of Line
x = 160
LCD

End of Line

Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


Window
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD

Window
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD

Window
Fetch
9C00
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD


Window
Fetch
9C00
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD


Window
Fetch
9C00
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD

Window
Fetch
9C00
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD


Window
Fetch
9C00
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD


Window
Fetch
9C00
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
Window
WX = 26
LCD


Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD


Sprites
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD


Sprites
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD


Sprites
Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD

Sprites
Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD


Sprites
Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD


Sprites
Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD

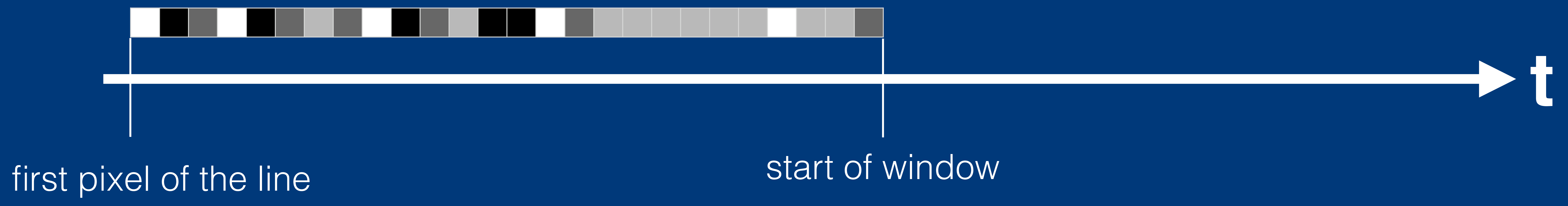
Sprites
Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD

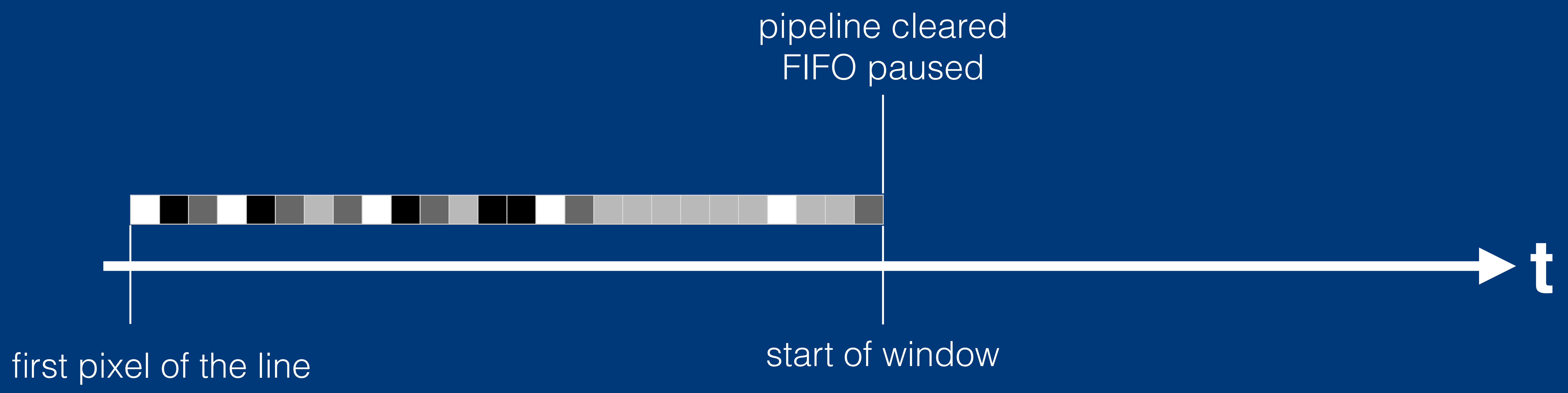

Sprites
Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD


Sprites
Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD

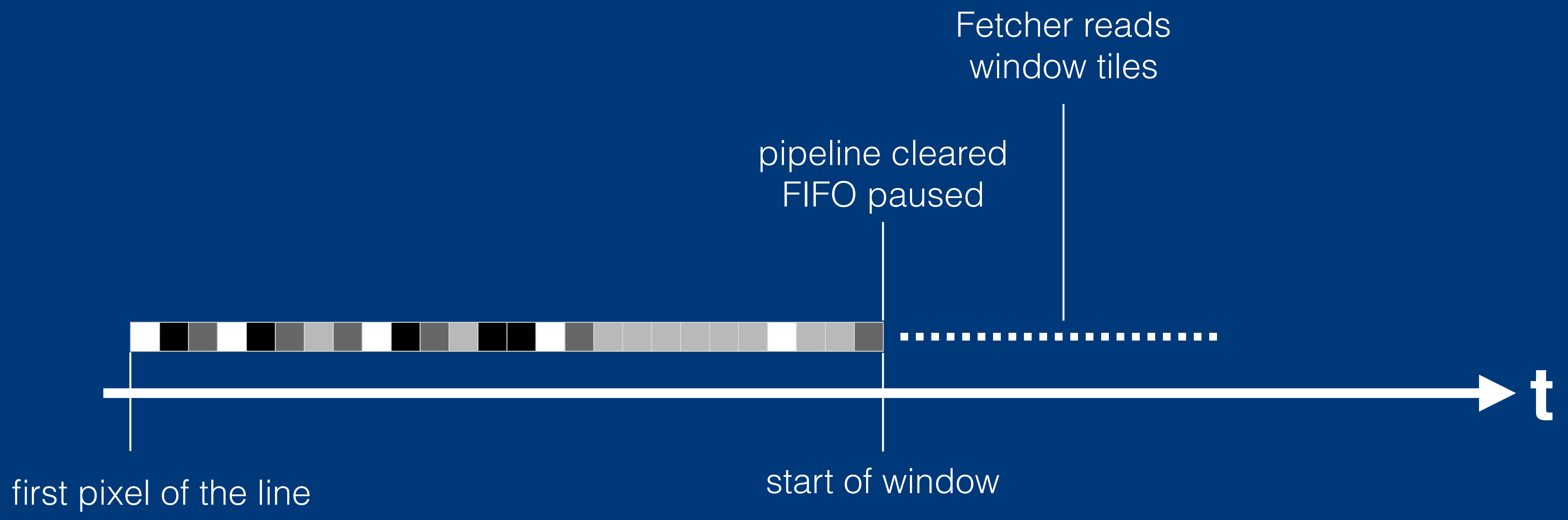

Sprites
Fetch
9804
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
OBJ
oam[0].x = 26
OBJ
oam[1].x = 40
LCD

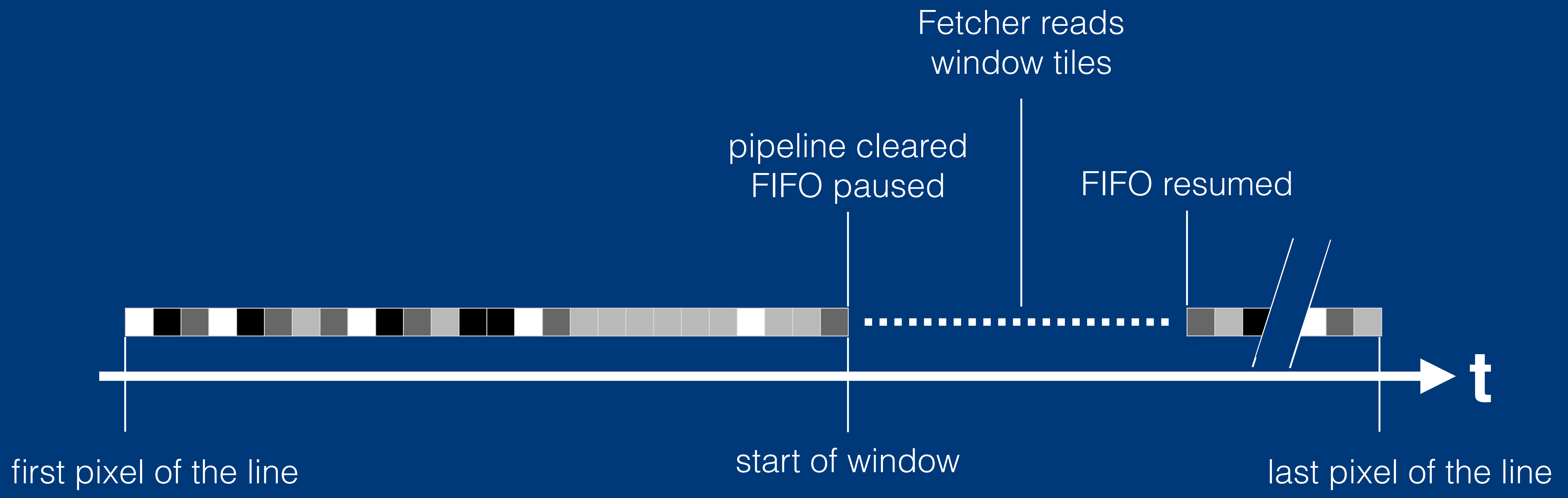

t

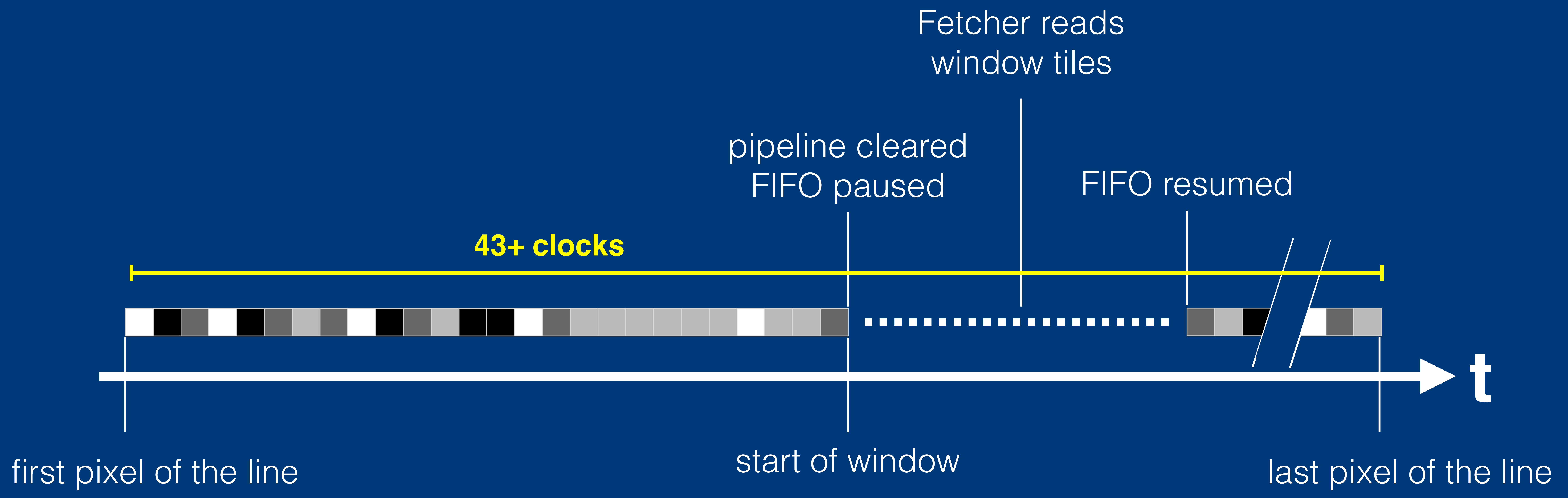

t
first pixel of the line
start of window



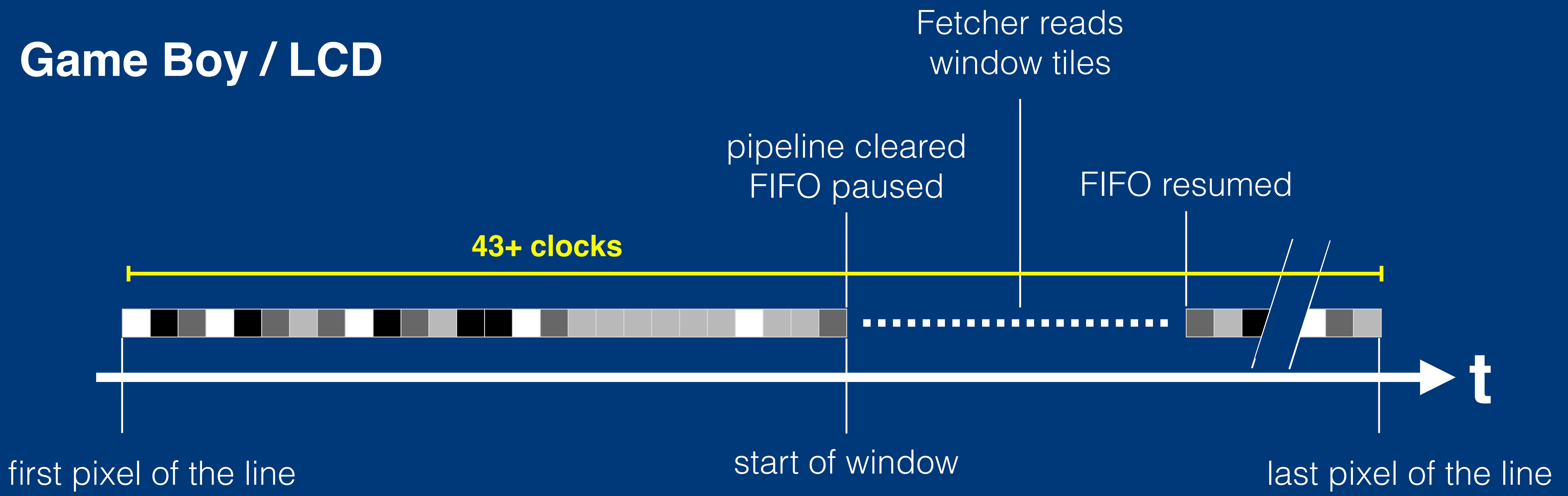
pipeline cleared
FIFO paused
t
first pixel of the line
start of window


Fetcher reads
window tiles
pipeline cleared
FIFO paused
t
first pixel of the line
start of window

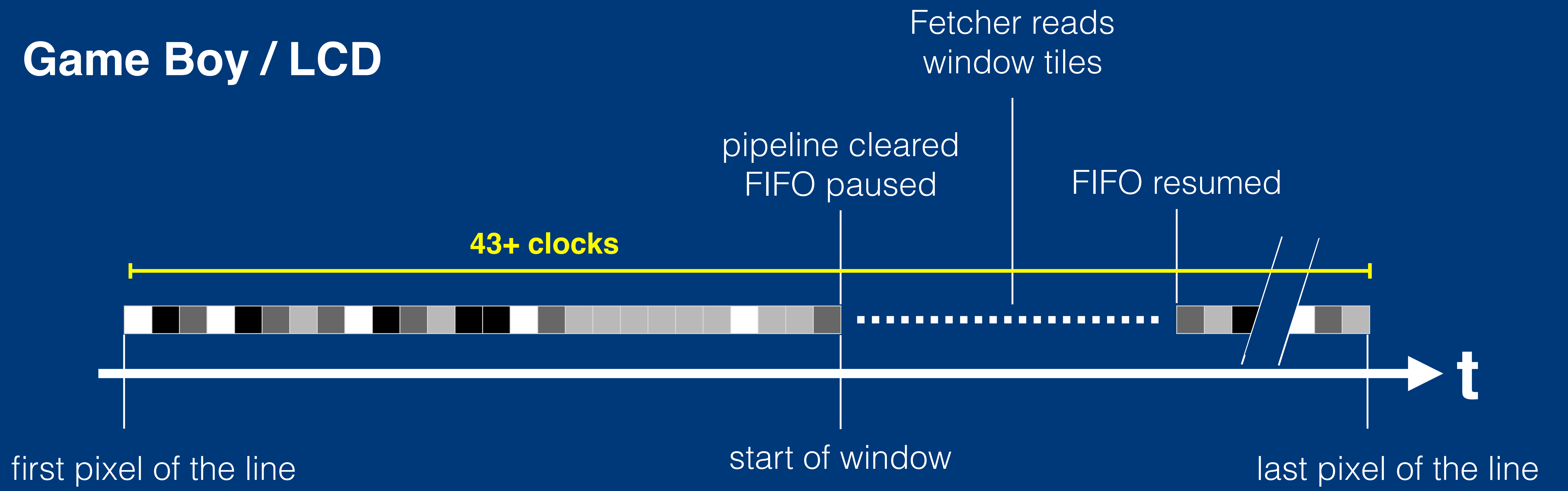

Fetcher reads
window tiles
pipeline cleared
FIFO paused
FIFO resumed
t
first pixel of the line
start of window
last pixel of the line


Fetcher reads
window tiles
pipeline cleared
FIFO paused
FIFO resumed
43+ clocks
t
first pixel of the line
start of window
last pixel of the line

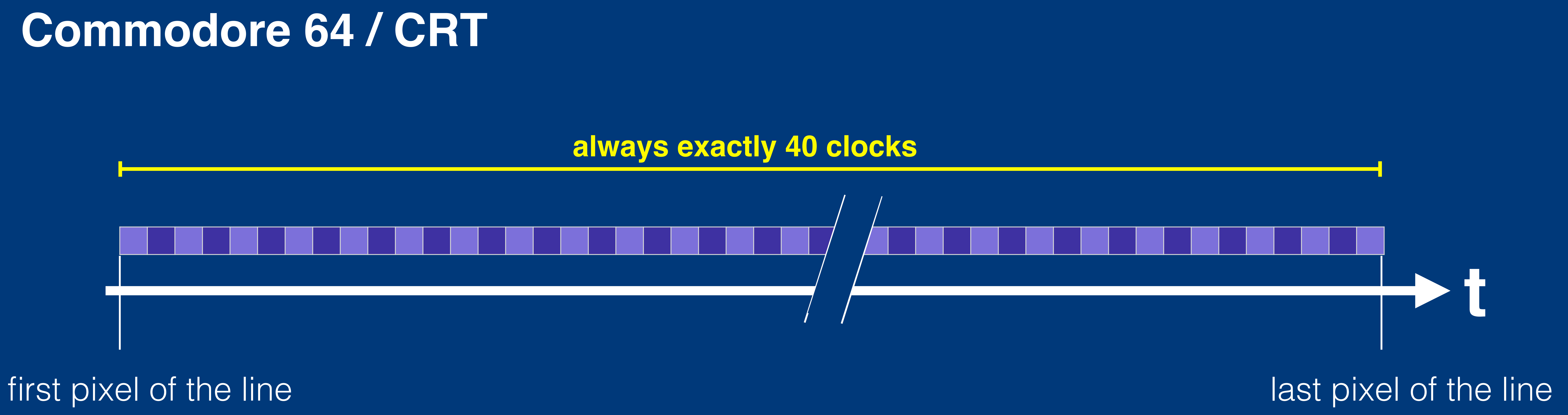
Game Boy / LCD



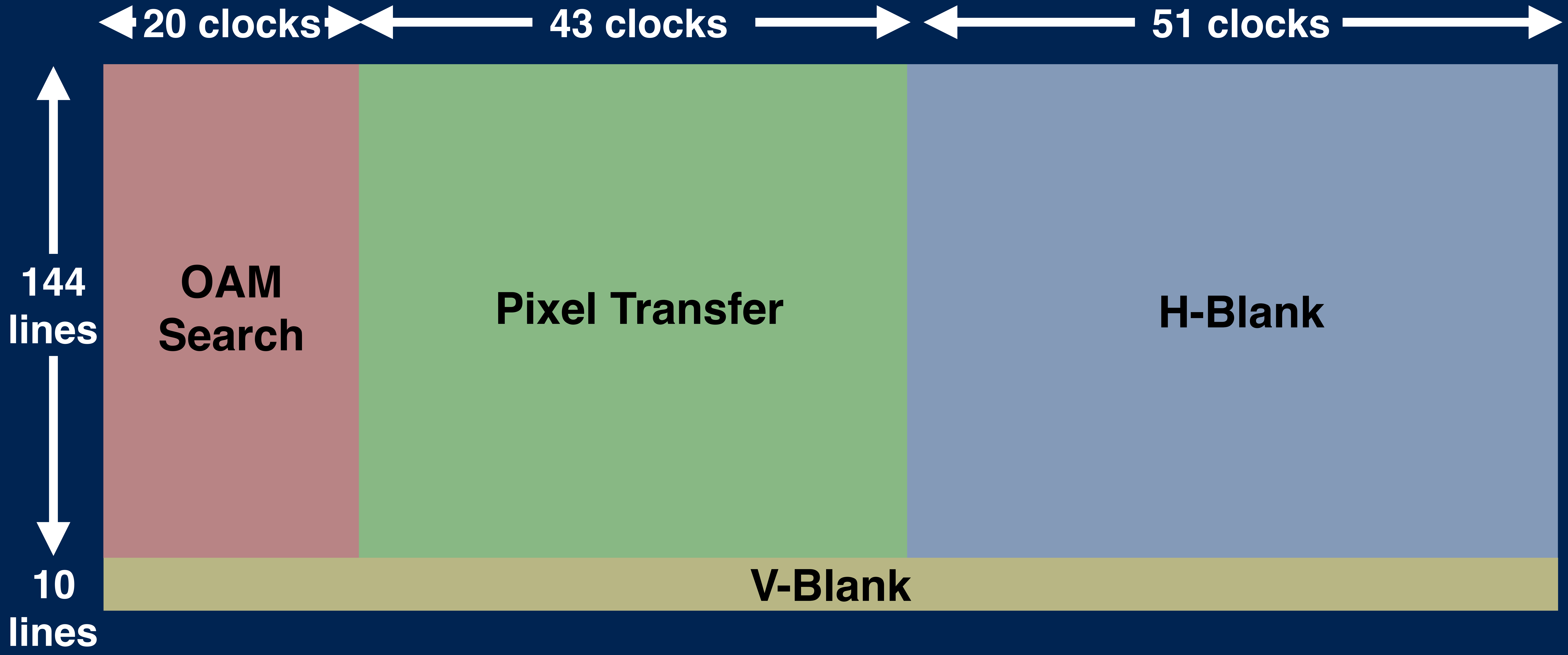
Commodore 64 / CRT

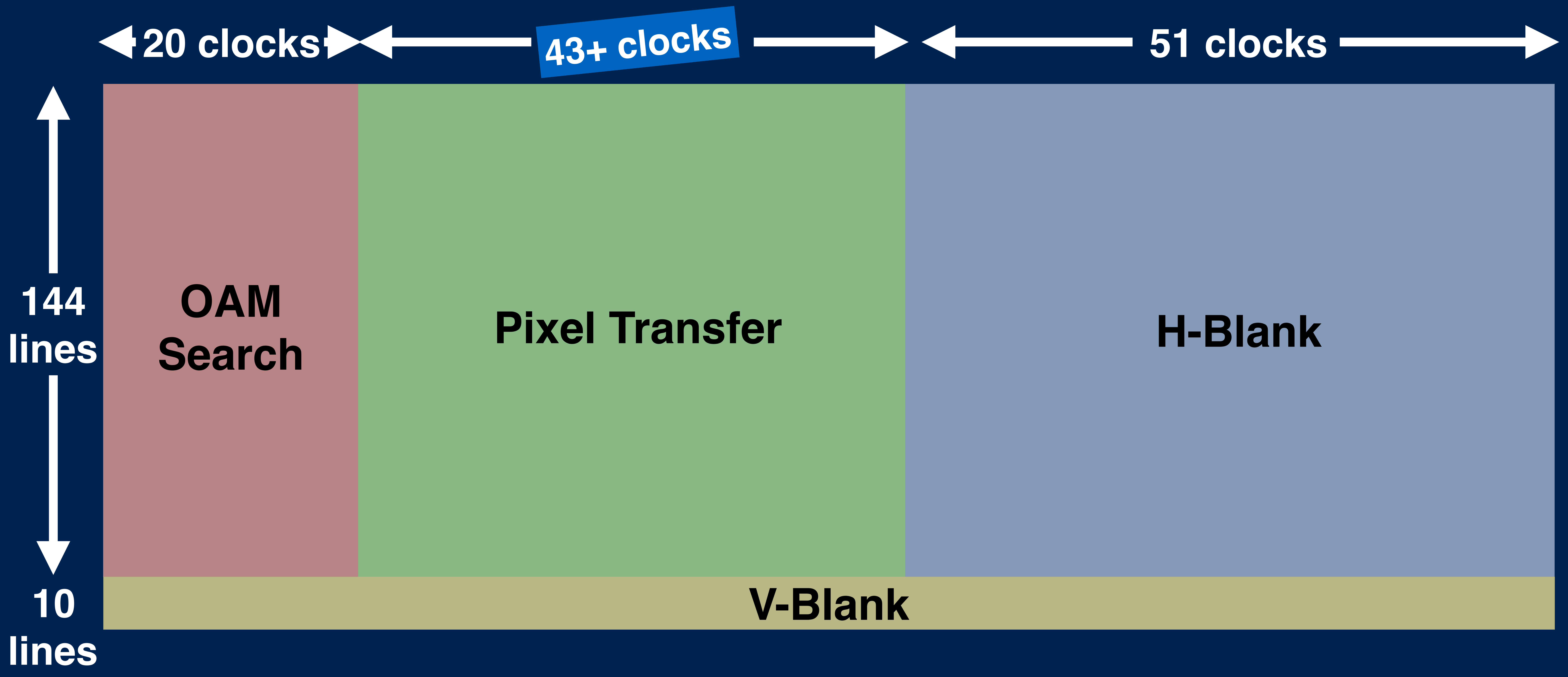


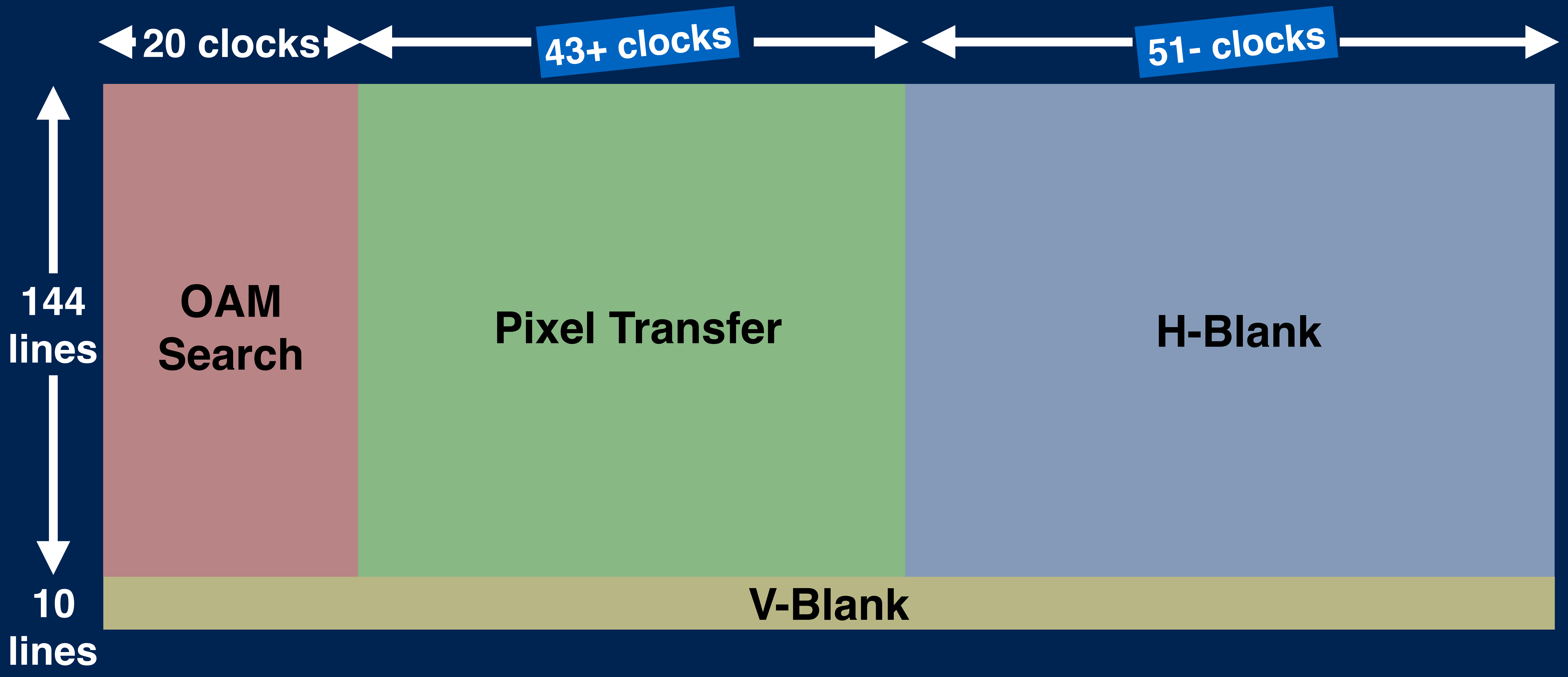
Game Boy / LCD
Fetcher reads
window tiles
pipeline cleared
FIFO paused
FIFO resumed
43+ clocks
t
first pixel of the line
start of window
last pixel of the line

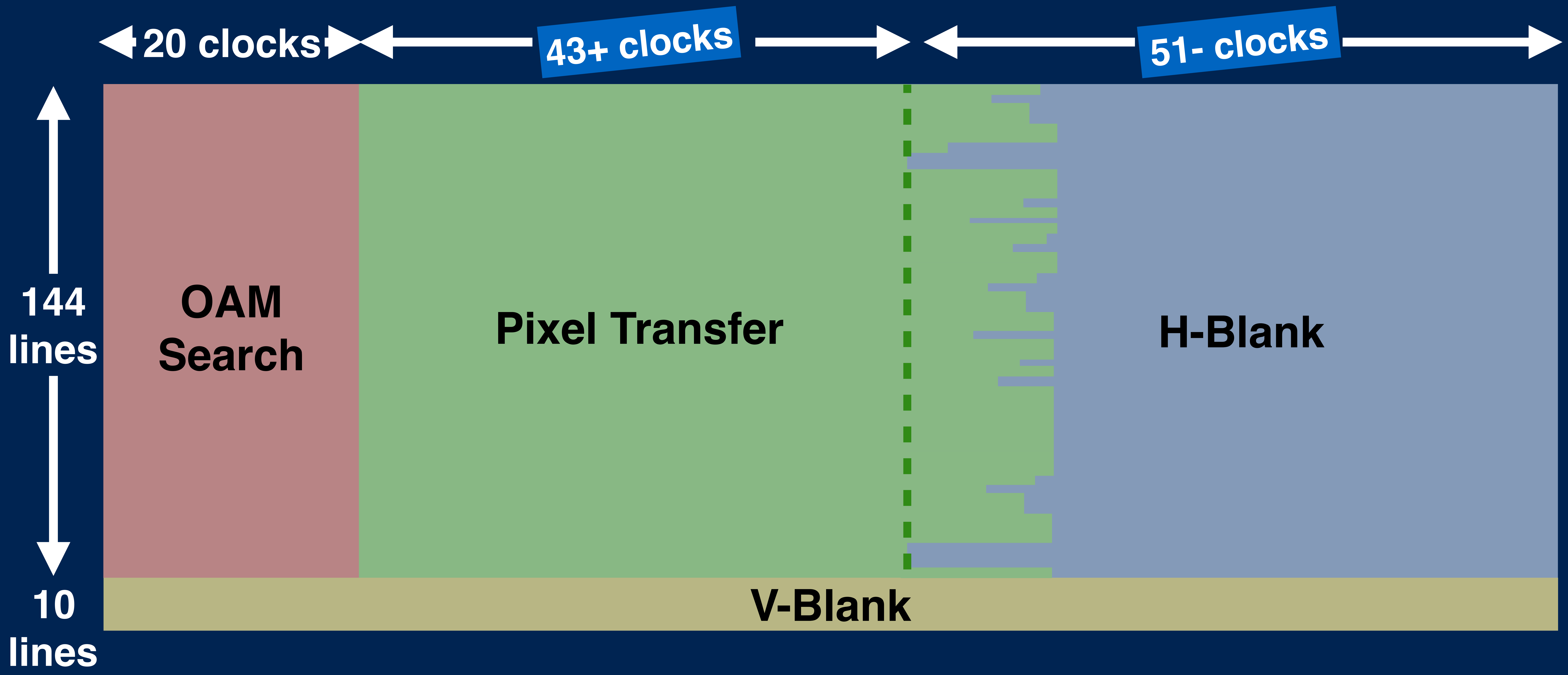


Commodore 64 / CRT
always exactly 40 clocks
t
first pixel of the line
last pixel of the line


20 clocks
43 clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
10 lines
V-Blank


20 clocks
43+ clocks
51 clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
10 lines
V-Blank


20 clocks
43+ clocks
51- clocks
144 lines
OAM Search
Pixel Transfer
H-Blank
10 lines
V-Blank


20 clocks
43+ clocks
51- clocks
144 lines
10 lines
OAM Search
Pixel Transfer
H-Blank
V-Blank


Fetch
FE02
Read Tile #
Read Data 0
Read Data 1
Pixel FIFO
LCD

Pixel Mixing

Pixel Mixing

Fetch

FE02

Read Tile #
Read Data 0
→Read Data 1

10	01	01	10	10	11	01	10	00							
BG	BG	BG	BG	BG	BG	BG	BG	BG							

Pixel FIFO

LCD

Pixel Mixing

Pixel Mixing

Pixel Mixing

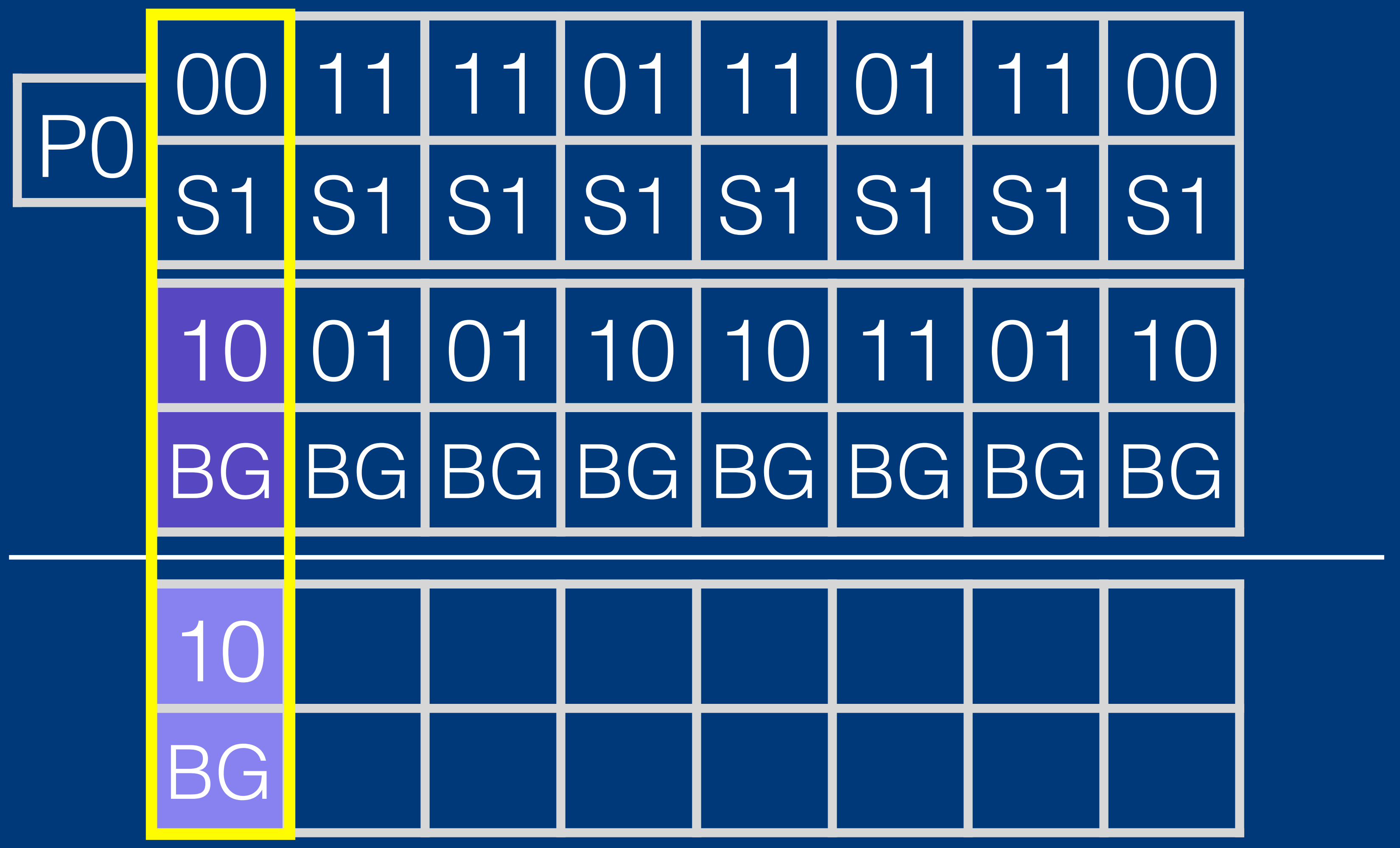
P0	00	11	11	01	11	01	11	00
	S1	S1	S1	S1	S1	S1	S1	S1

10	01	01	10	10	11	01	10
BG	BG	BG	BG	BG	BG	BG	BG

Pixel Mixing

P0	00	11	11	01	11	01	11	00
	S1	S1	S1	S1	S1	S1	S1	S1
	10	01	01	10	10	11	01	10
	BG	BG	BG	BG	BG	BG	BG	BG

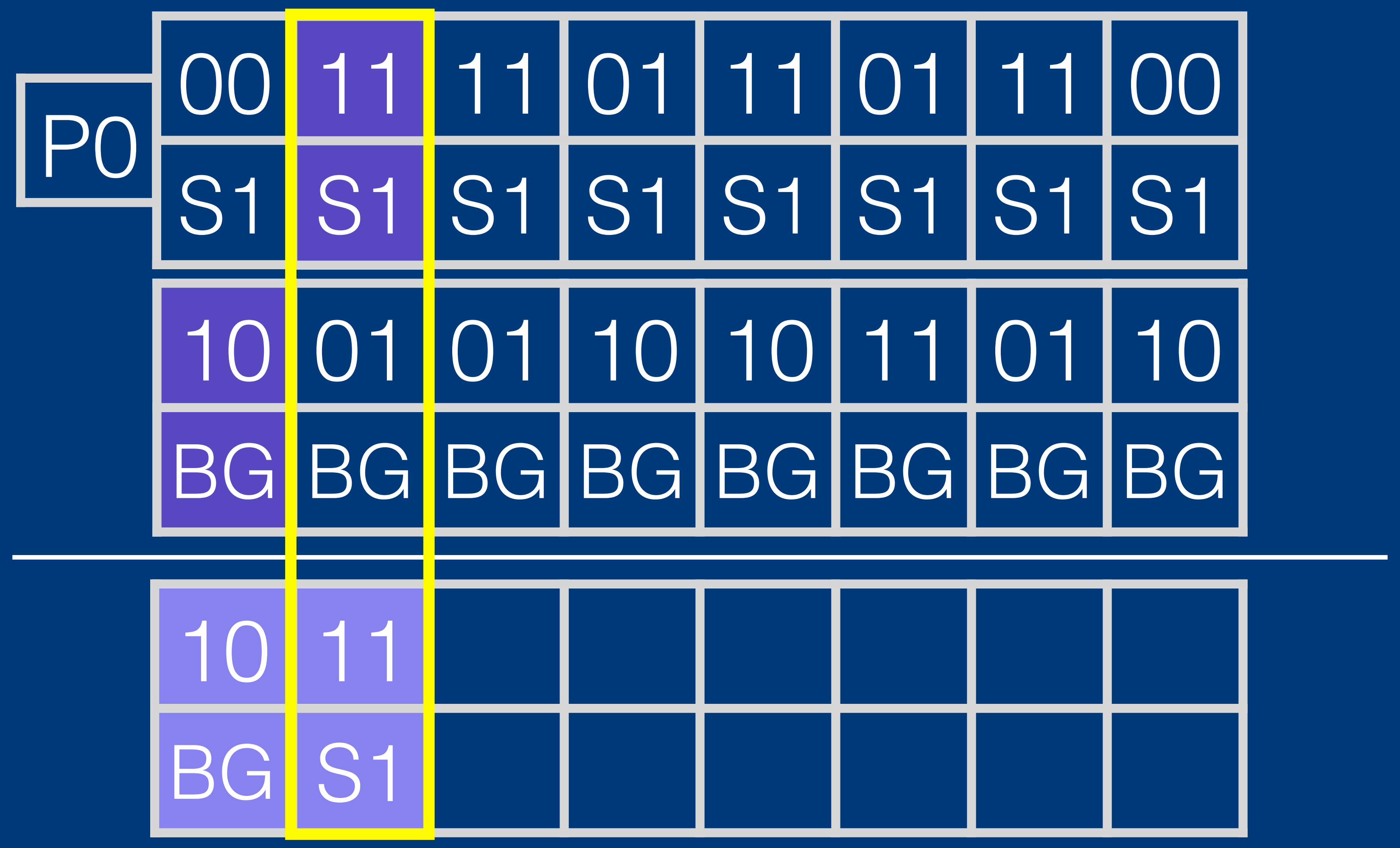
Pixel Mixing



Pixel Mixing

P0	00	11	11	01	11	01	11	00
	S1	S1	S1	S1	S1	S1	S1	S1
	10	01	01	10	10	11	01	10
	BG	BG	BG	BG	BG	BG	BG	BG
	10							
	BG							

Pixel Mixing



Pixel Mixing

P0	00	11	11	01	11	01	11	00
	S1	S1	S1	S1	S1	S1	S1	S1
	10	01	01	10	10	11	01	10
	BG	BG	BG	BG	BG	BG	BG	BG
	10	11	11	01	11	01	11	
	BG	S1	S1	S1	S1	S1	S1	

Pixel Mixing

P0	00	11	11	01	11	01	11	00
	S1	S1	S1	S1	S1	S1	S1	S1
	10	01	01	10	10	11	01	10
	BG	BG	BG	BG	BG	BG	BG	BG
	10	11	11	01	11	01	11	10
	BG	S1	S1	S1	S1	S1	S1	BG

Pixel Mixing

P0	00	11	11	01	11	01	11	00
	S1	S1	S1	S1	S1	S1	S1	S1
	10	01	01	10	10	11	01	10
	BG	BG	BG	BG	BG	BG	BG	BG
	10	11	11	01	11	01	11	10
	BG	S1	S1	S1	S1	S1	S1	BG

Pixel Mixing

10	11	11	01	11	01	11	10
BG	S1	S1	S1	S1	S1	S1	BG

Pixel Mixing

P0	01	01	01	01	01	01	01	01
	S0	S0	S0	S0	S0	S0	S0	S0

10	11	11	01	11	01	11	10
BG	S1	S1	S1	S1	S1	S1	BG

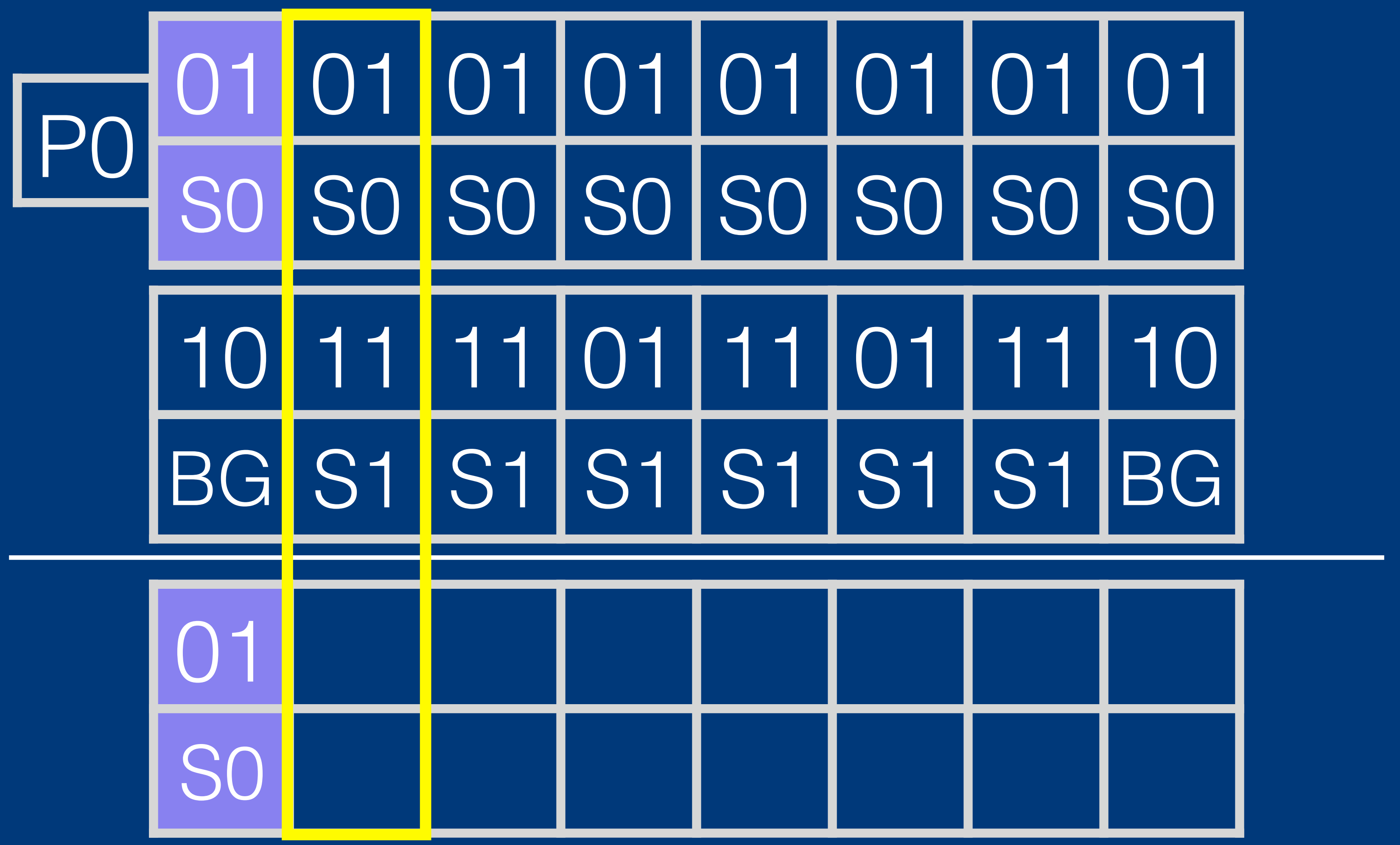
Pixel Mixing

P0	01	01	01	01	01	01	01	01
	S0	S0	S0	S0	S0	S0	S0	S0
	10	11	11	01	11	01	11	10
	BG	S1	S1	S1	S1	S1	S1	BG

Pixel Mixing

P0	01	01	01	01	01	01	01	01
	S0	S0	S0	S0	S0	S0	S0	S0
	10	11	11	01	11	01	11	10
	BG	S1	S1	S1	S1	S1	S1	BG
	01							
	S0							

Pixel Mixing



Pixel Mixing

P0	01	01	01	01	01	01	01	01
	S0	S0	S0	S0	S0	S0	S0	S0
	10	11	11	01	11	01	11	10
	BG	S1	S1	S1	S1	S1	S1	BG
	01	11						
	S0	S1						

Pixel Mixing

Pixel Mixing

P0	01	01	01	01	01	01	01	01
	S0	S0	S0	S0	S0	S0	S0	S0
	10	11	11	01	11	01	11	10
	BG	S1	S1	S1	S1	S1	S1	BG
	01	11	11	01	11	01	11	01
	S0	S1	S1	S1	S1	S1	S1	S0

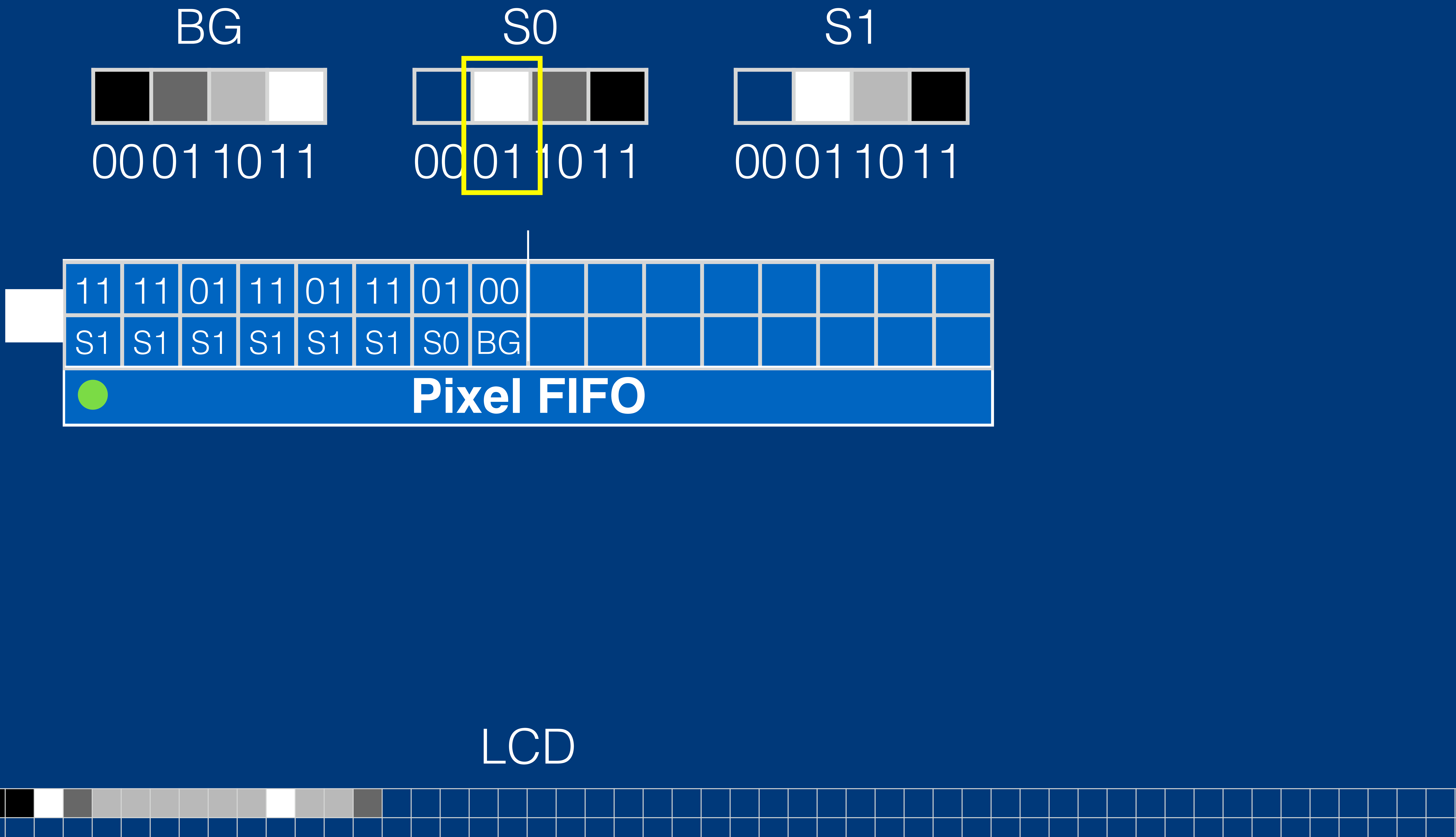

01 11 11 01 11 01 11 01 00
S0 S1 S1 S1 S1 S1 S1 S0 BG
Pixel FIFO
LCD

Applying the Palettes

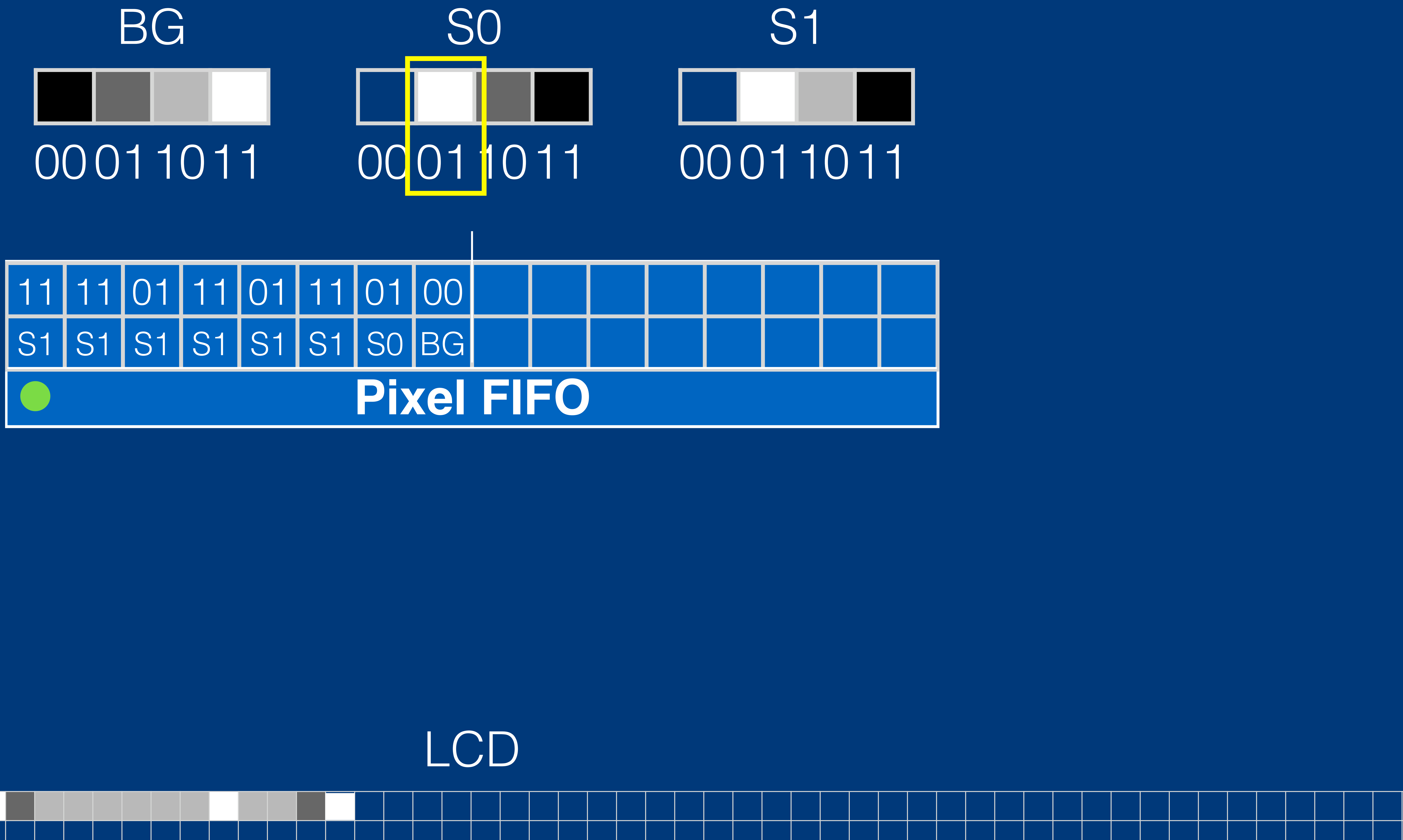
Applying the Palettes

Applying the Palettes

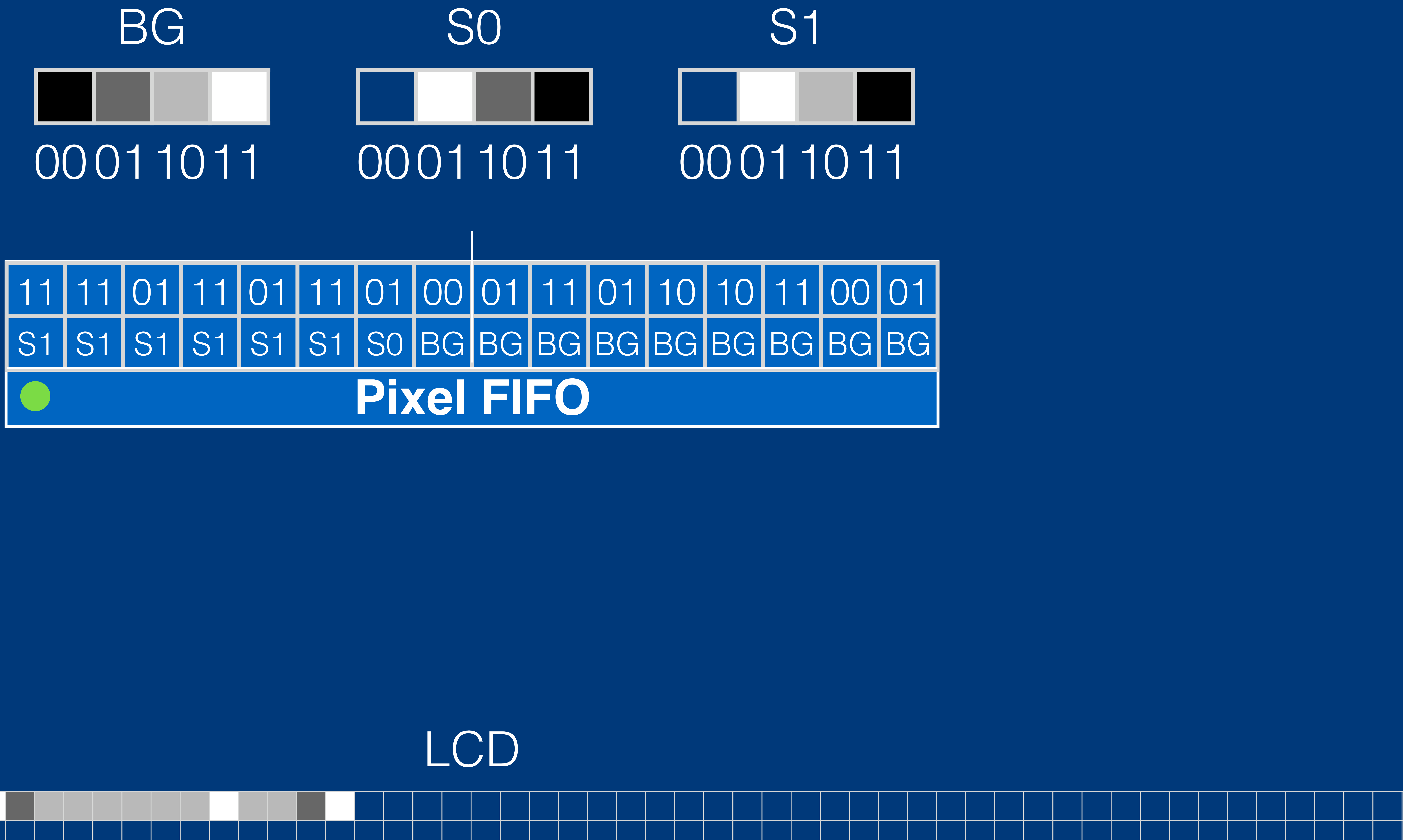
Applying the Palettes



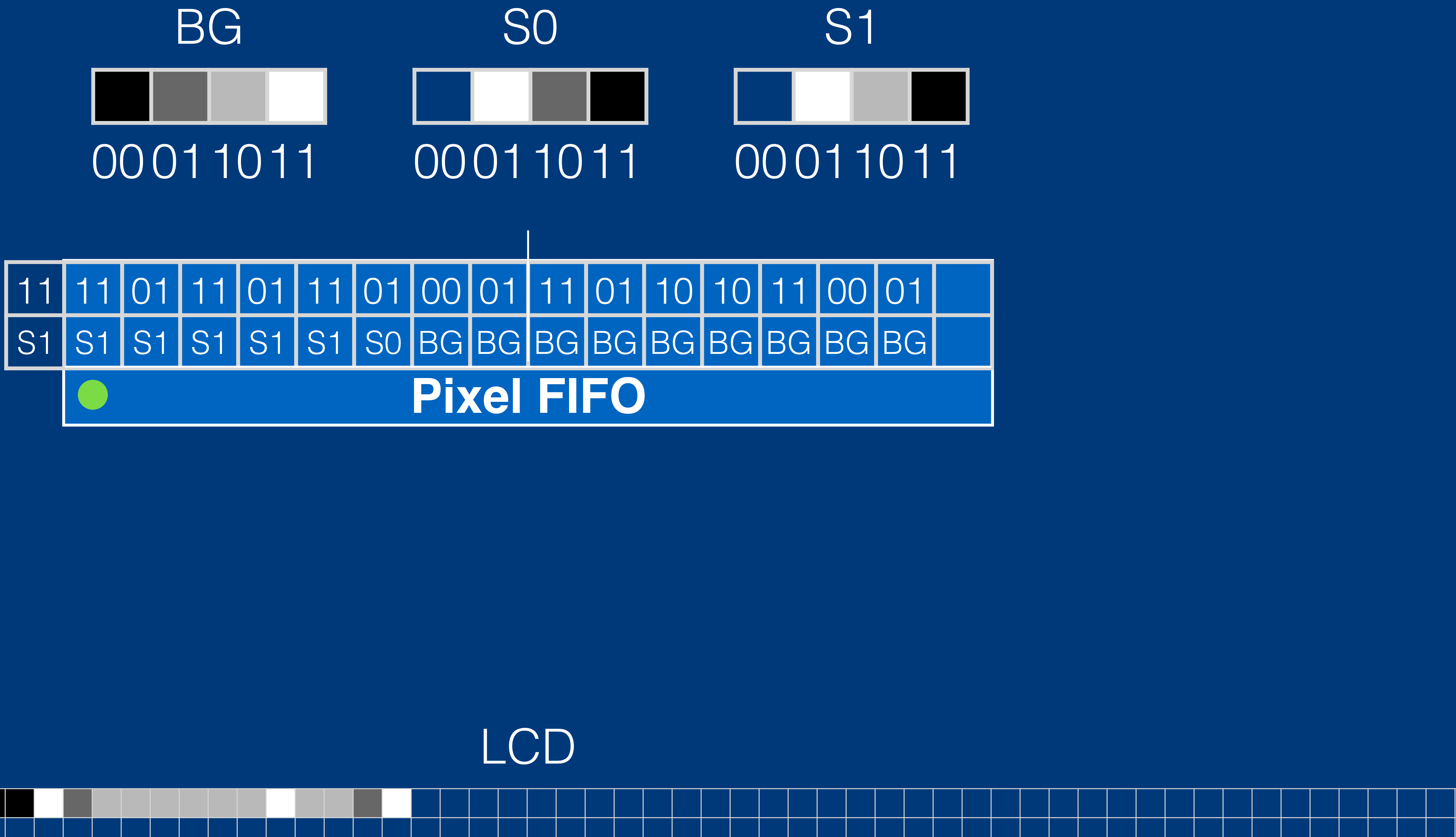
Applying the Palettes



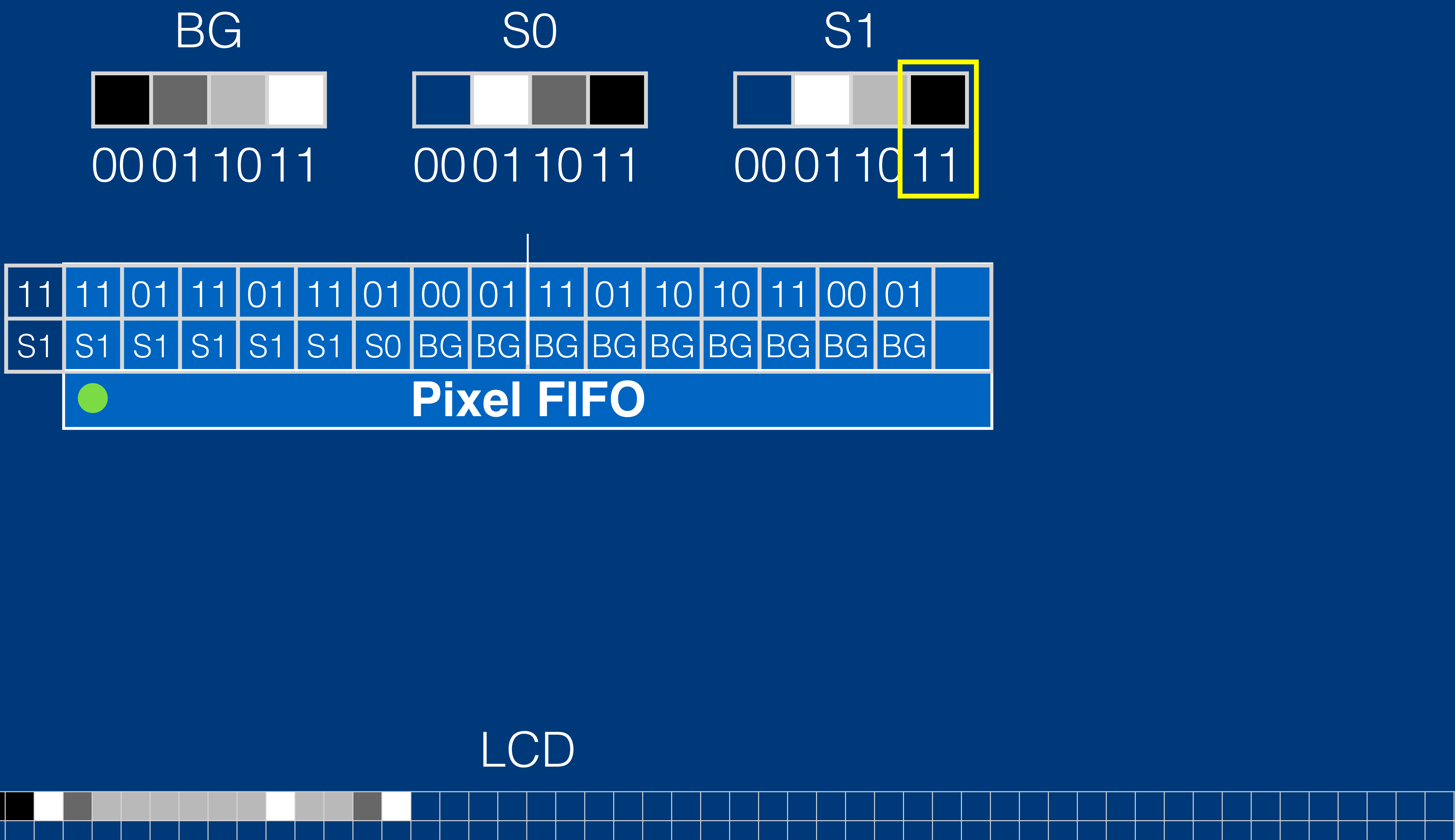
Applying the Palettes



Applying the Palettes



Applying the Palettes

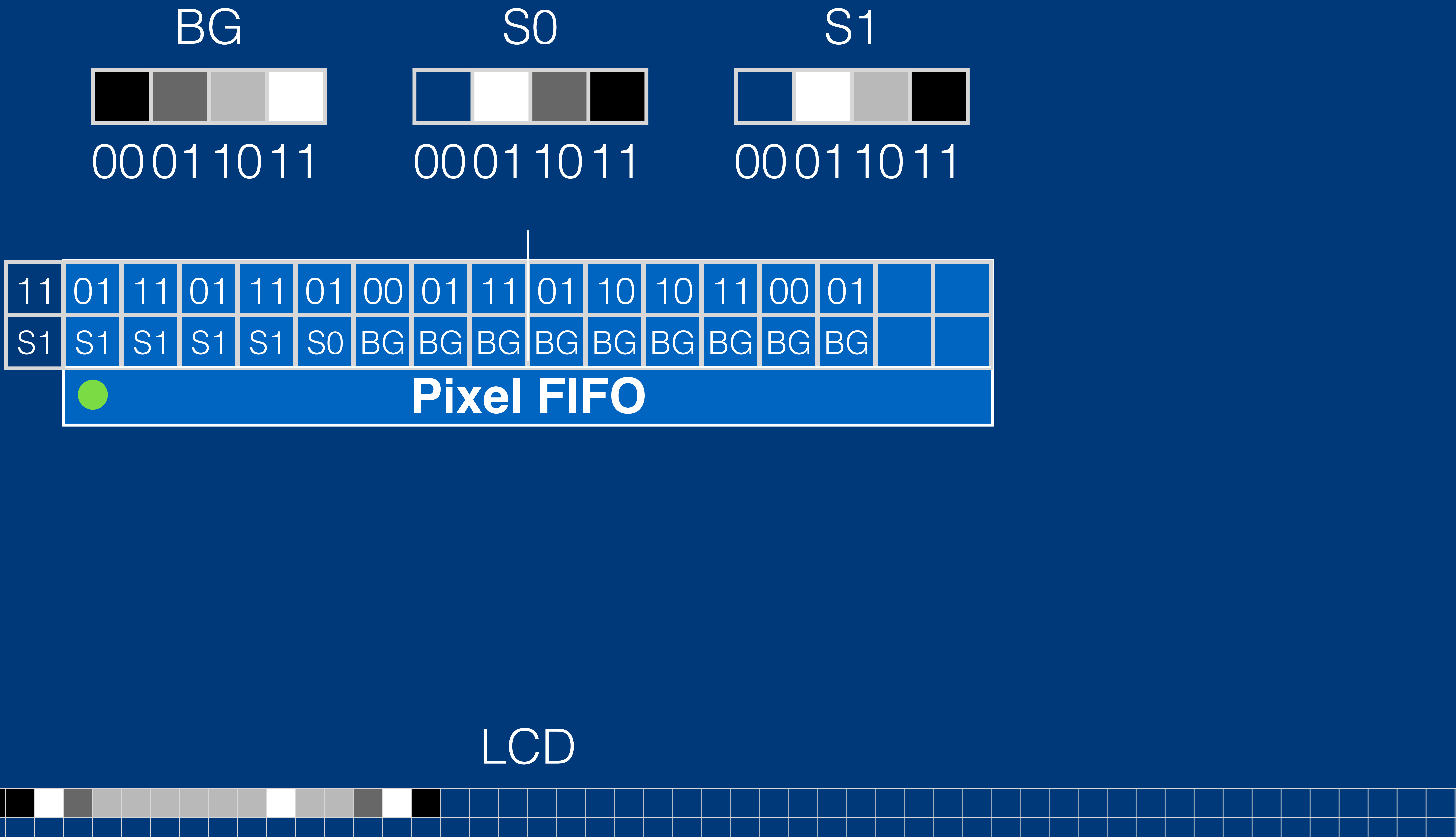


Applying the Palettes

Applying the Palettes

Applying the Palettes

Applying the Palettes



Applying the Palettes

Applying the Palettes

Applying the Palettes


BG
S0
S1
00 01 10 11
00 01 10 11
00 01 10 11
01 11 01 11 01 00 01 11 01 10 10 11 00 01
S1 S1 S1 S1 S0 BG BG BG BG BG BG BG BG BG
Pixel FIFO
LCD

Colorization

BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

01	11	01	11	01	00	01	11	01	10	10	11	00	01		
S1	S1	S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG		

Pixel FIFO

LCD

Colorization

BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

01	11	01	11	01	00	01	11	01	10	10	11	00	01		
S1	S1	S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG		

Pixel FIFO

LCD

Colorization

BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

01	11	01	11	01	00	01	11	01	10	10	11	00	01		
S1	S1	S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG		

Pixel FIFO

LCD

Colorization

BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

01	11	01	11	01	00	01	11	01	10	10	11	00	01			
S1	S1	S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG			

Pixel FIFO

LCD

Colorization

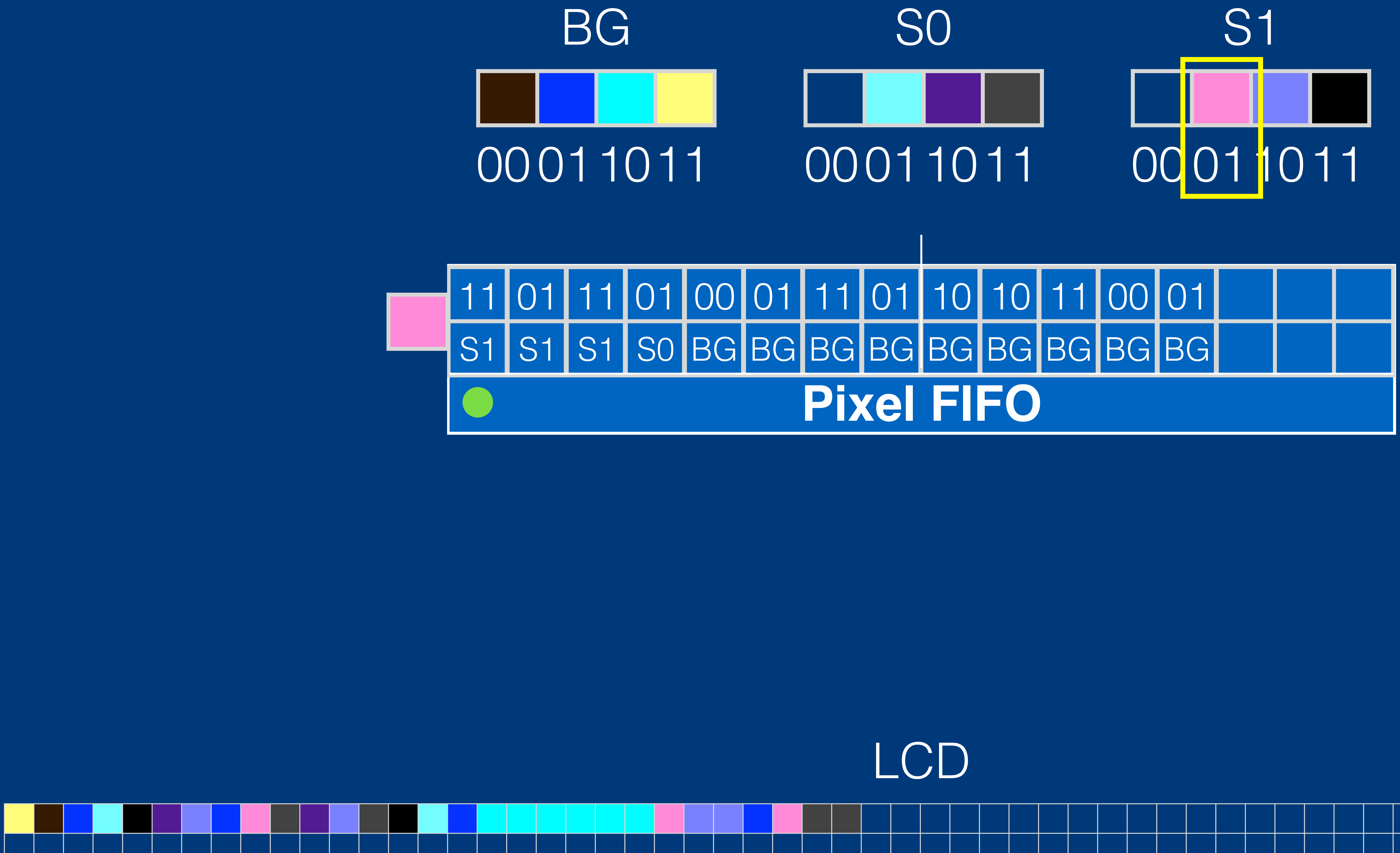
BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

01	11	01	11	01	00	01	11	01	10	10	11	00	01			
S1	S1	S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG			

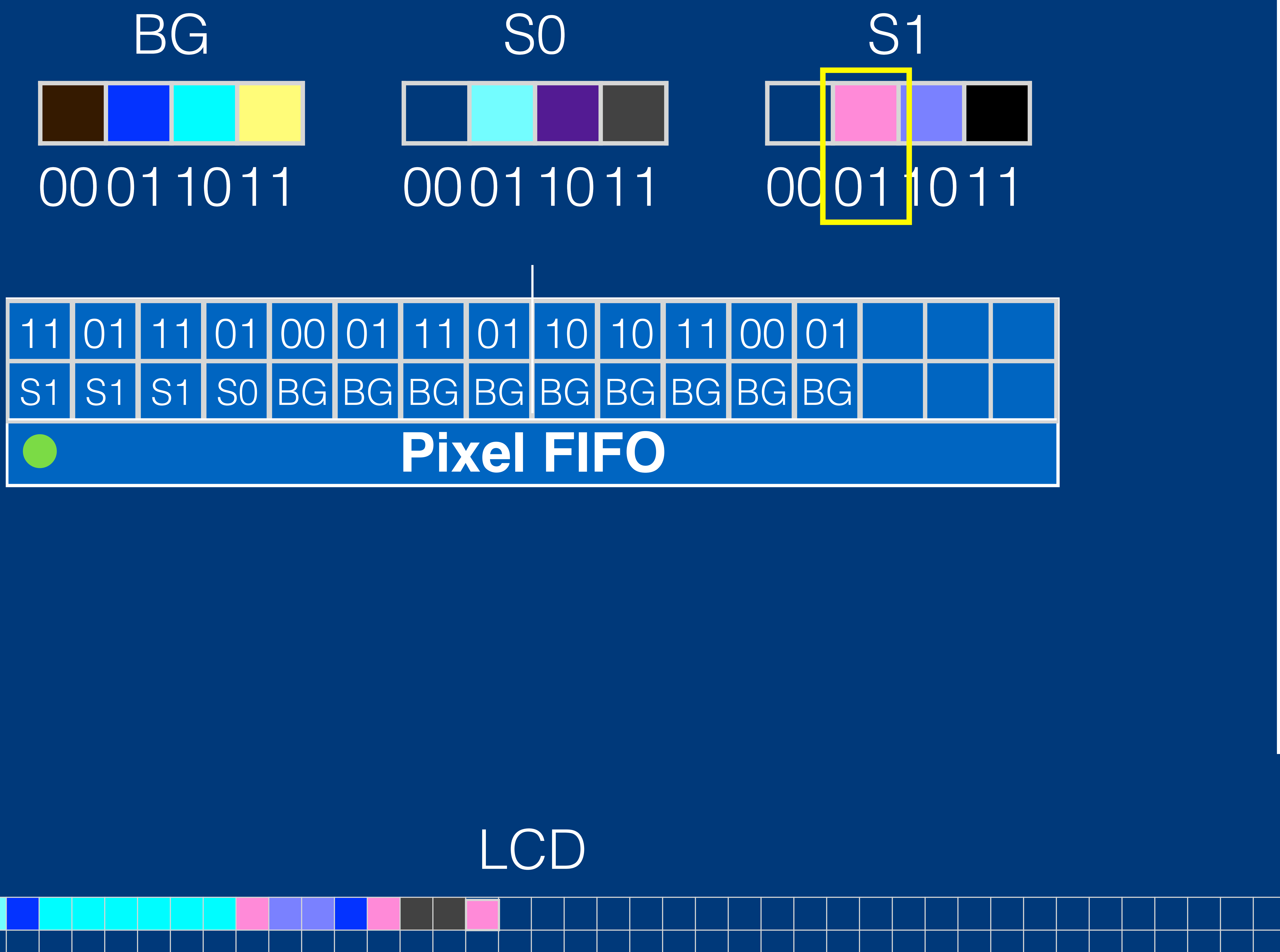
Pixel FIFO

LCD

Colorization



Colorization



Colorization

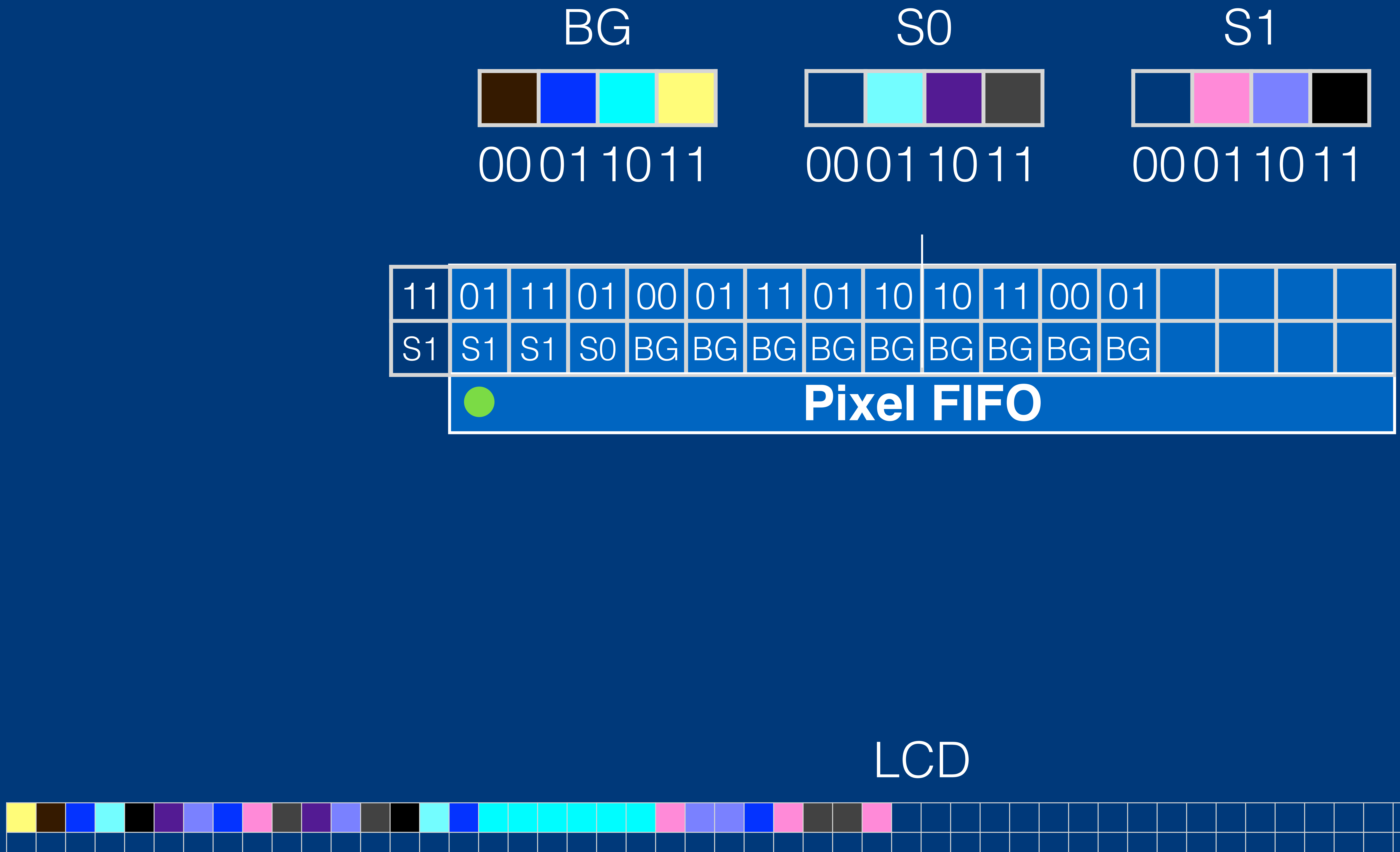
BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

11	01	11	01	00	01	11	01	10	10	11	00	01			
S1	S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG			

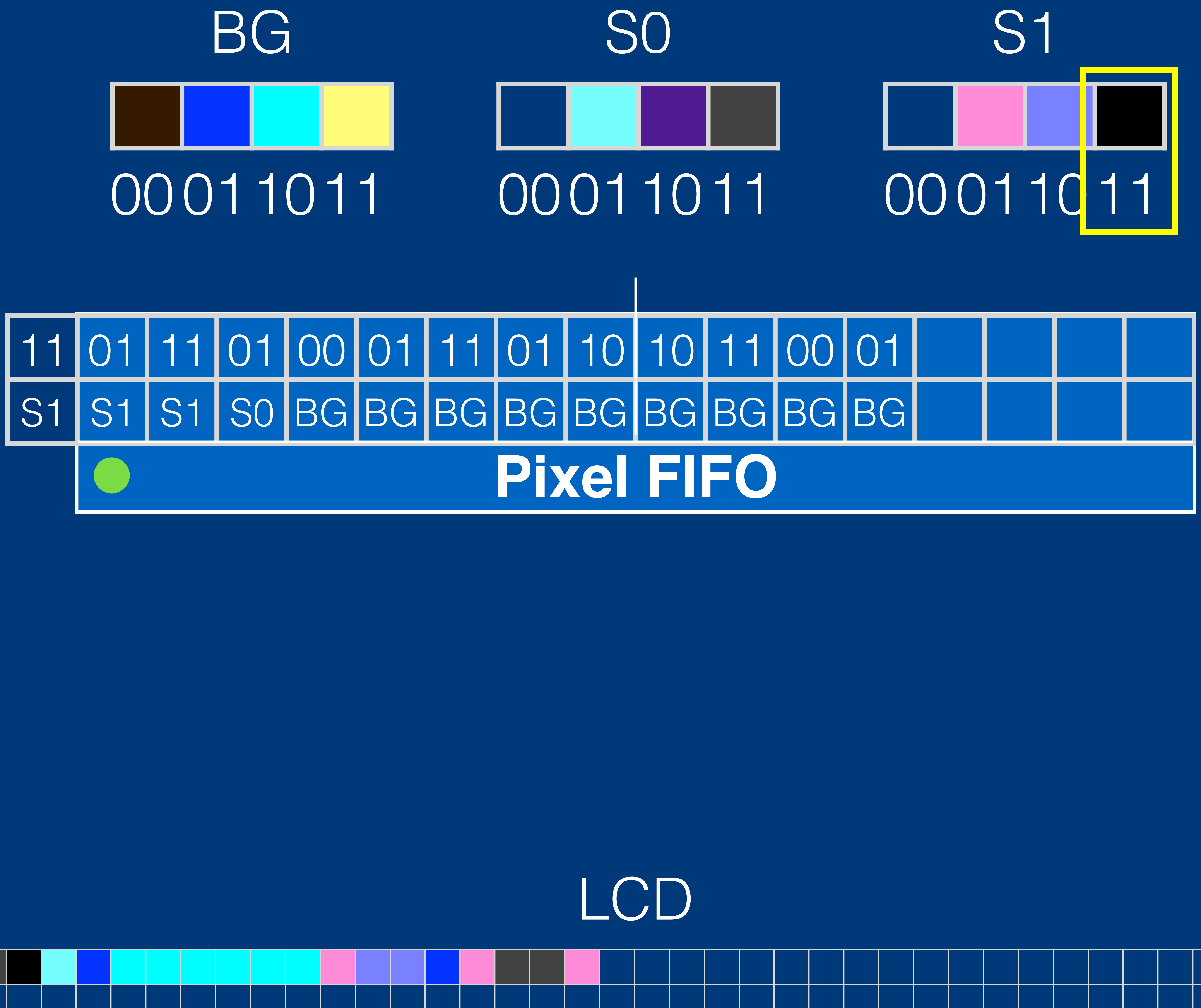
Pixel FIFO

LCD

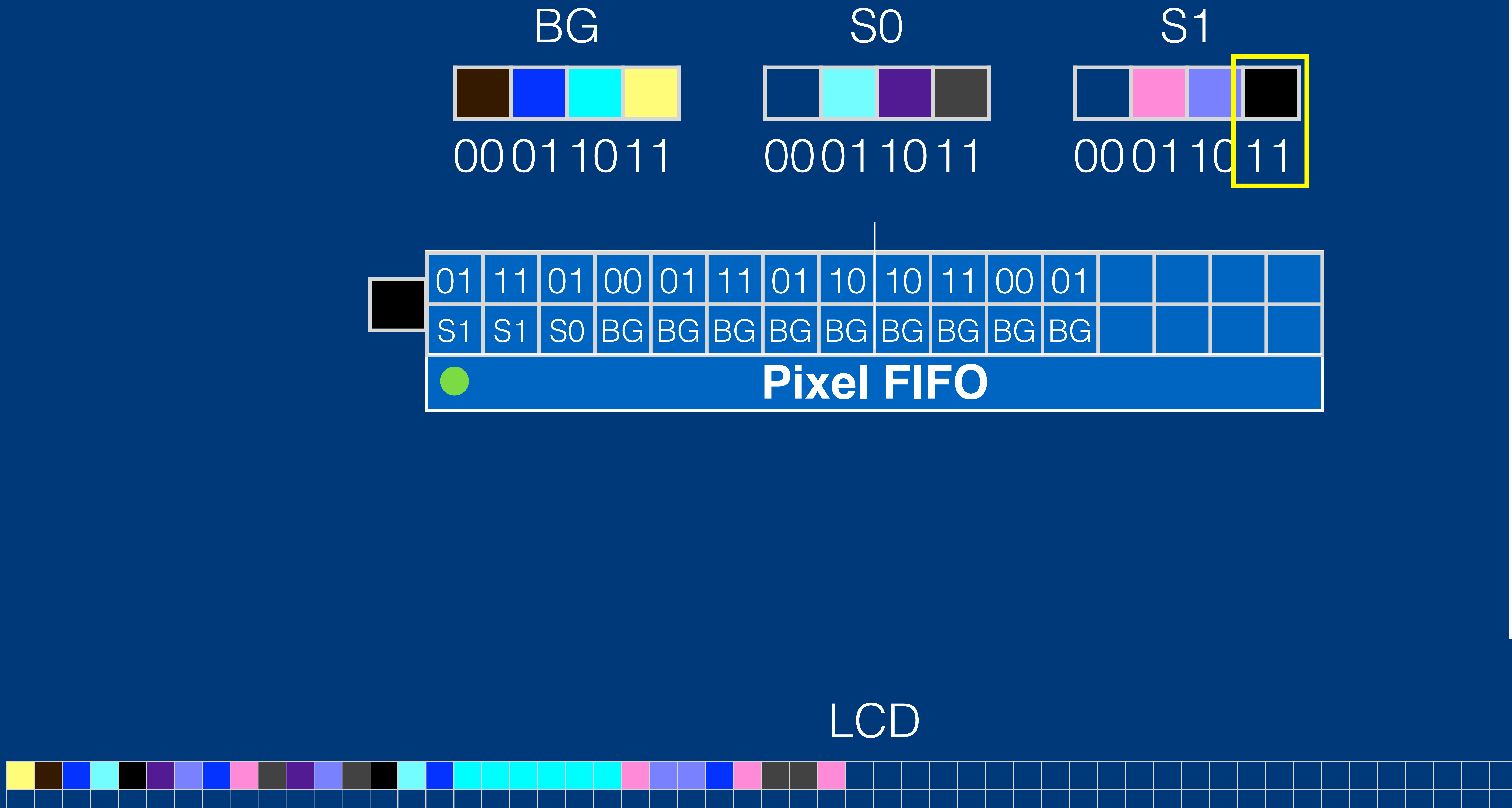
Colorization



Colorization



Colorization



Colorization

BG				S0				S1			
00	01	10	11	00	01	10	11	00	01	10	11

01	11	01	00	01	11	01	10	10	11	00	01				
S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG				

Pixel FIFO

LCD

Colorization

BG

00 01 10 11

S0

00 01 10 11

S1

00 01 10 11

01	11	01	00	01	11	01	10	10	11	00	01				
S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG				

Pixel FIFO

LCD

Colorization

BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

01	11	01	00	01	11	01	10	10	11	00	01					
S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG					

Pixel FIFO

LCD

Colorization

BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

01	11	01	00	01	11	01	10	10	11	00	01					
S1	S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG					

Pixel FIFO

LCD

Colorization

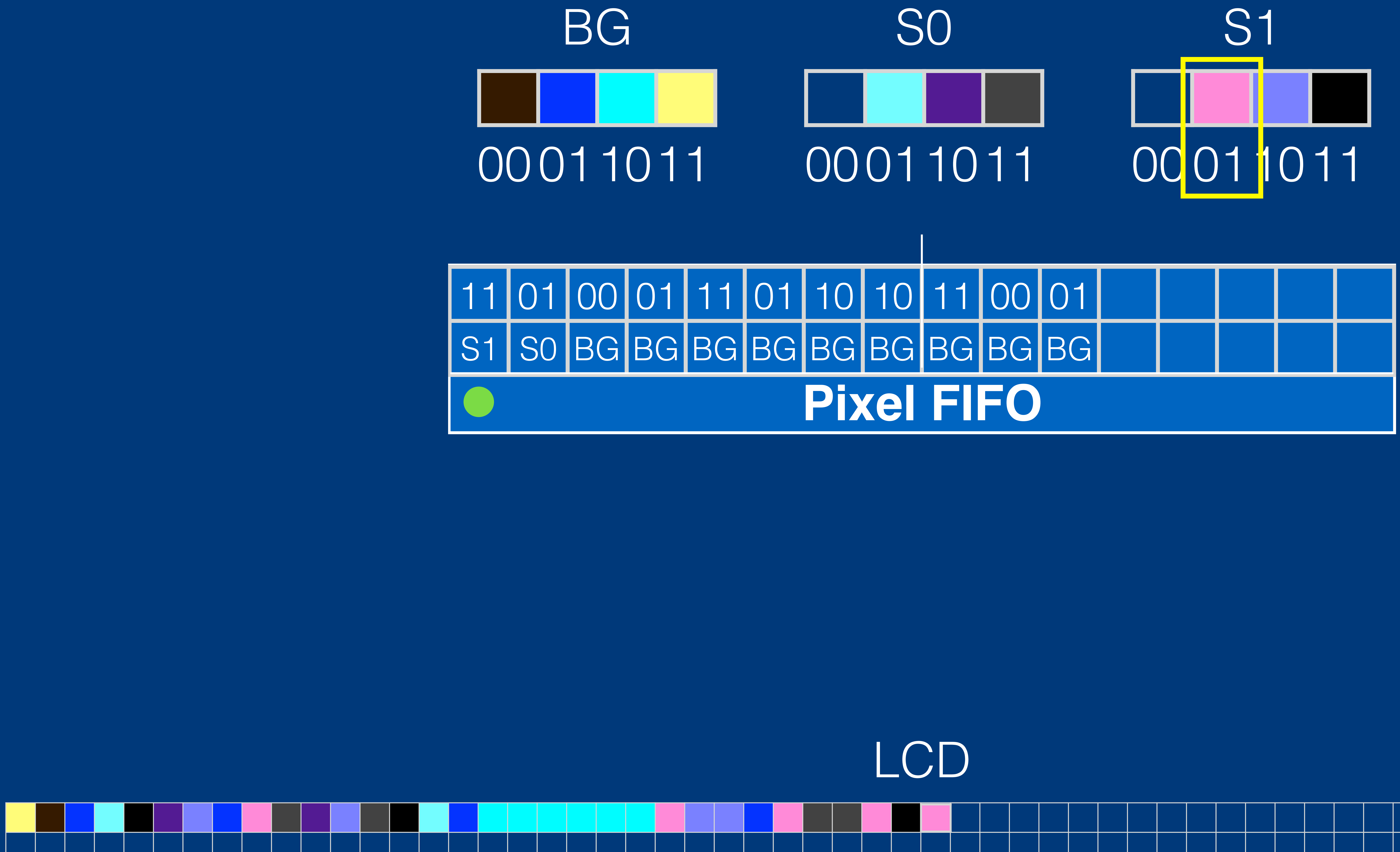
BG	S0	S1
00 01 10 11	00 01 10 11	00 01 10 11

11	01	00	01	11	01	10	10	11	00	01					
S1	S0	BG	BG	BG	BG	BG	BG	BG	BG	BG					

Pixel FIFO

LCD

Colorization



End of the technical part

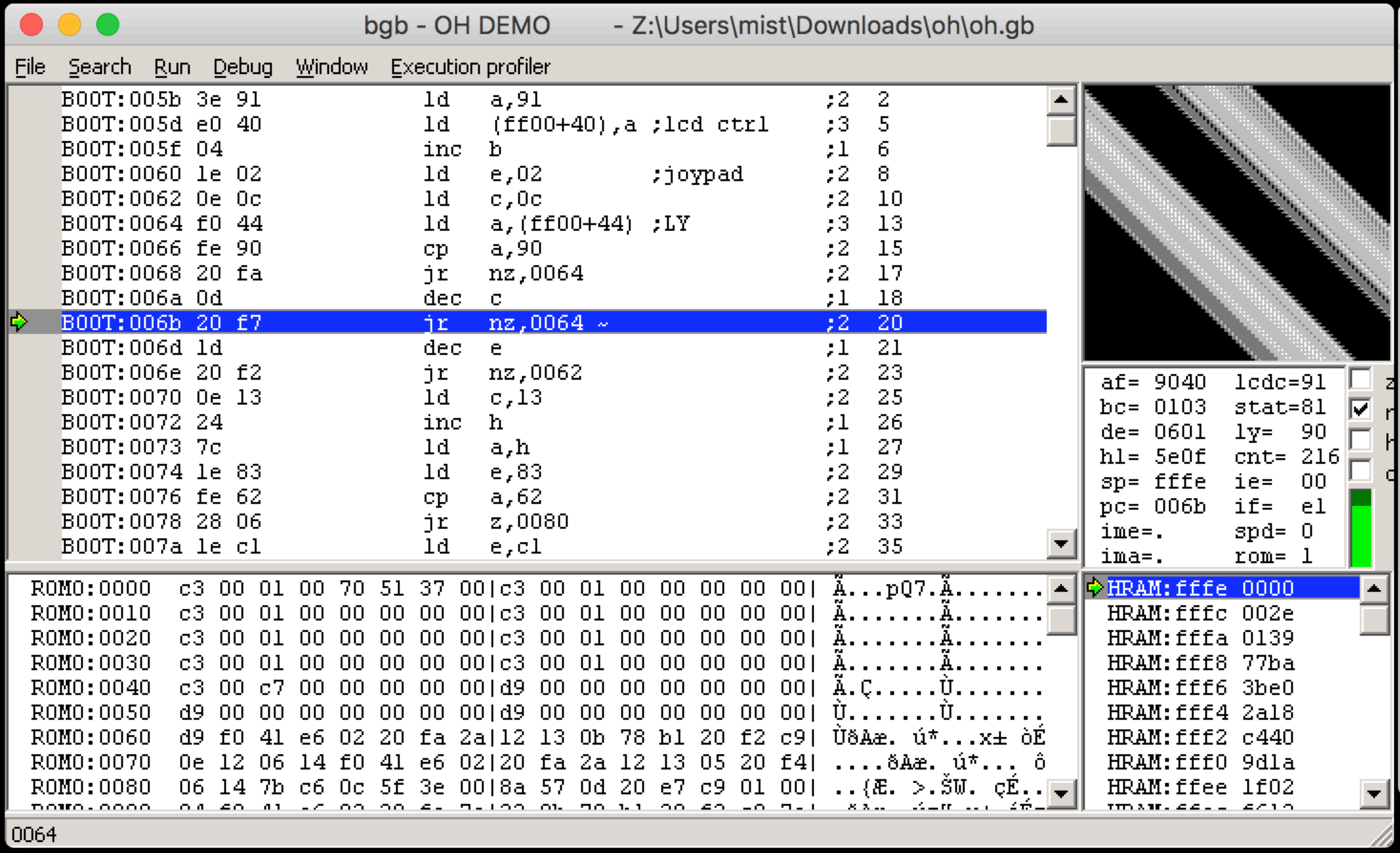
Development

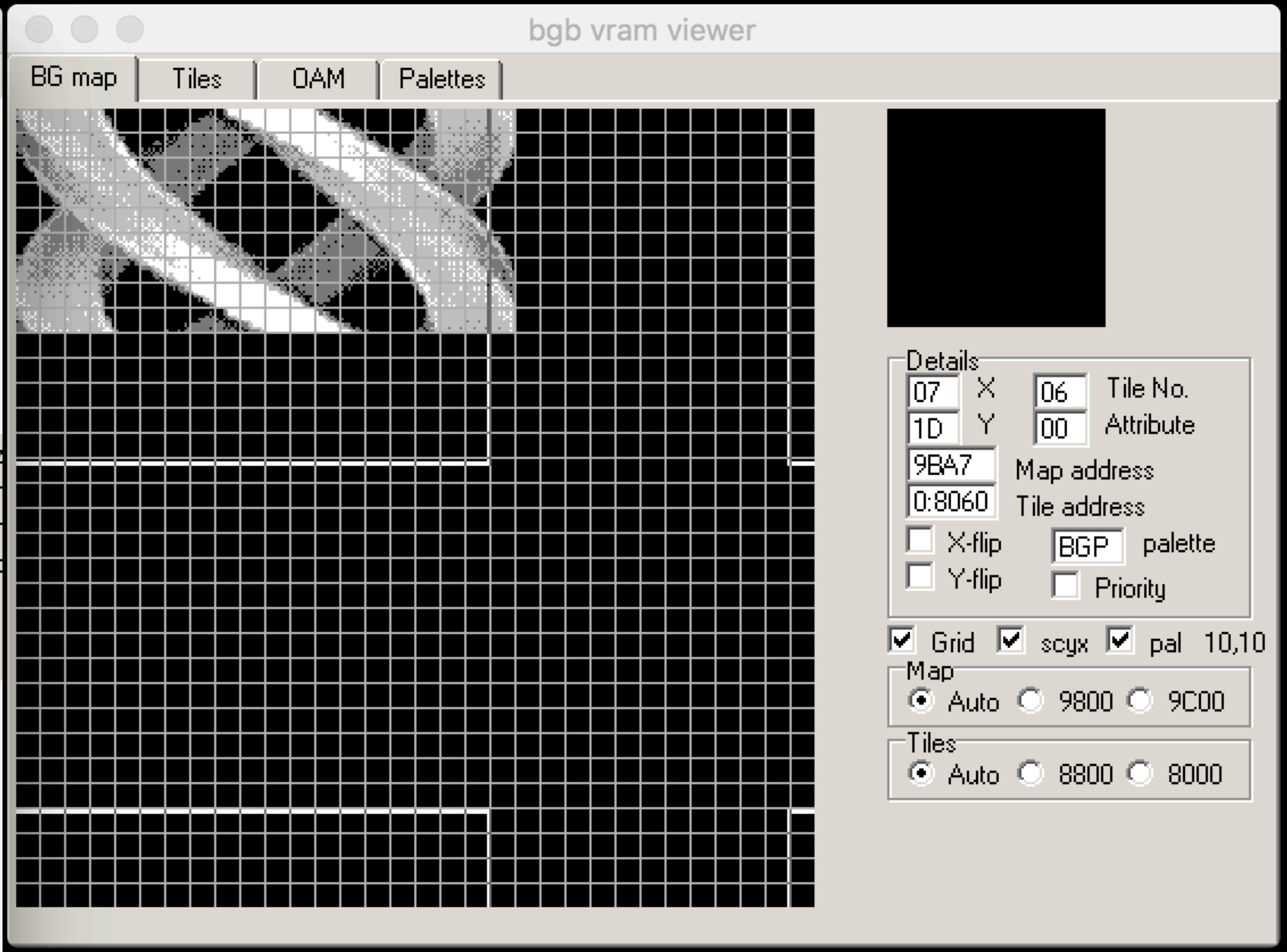
Rednex Game Boy Development System

```
$ rgbasm
Usage: rgbasm [-hv] [-b chars] [-Dname[=value]] [-g chars] [-i path]
              [-o outfile] [-p pad_value] file.asm
$ rgblink
usage: rgblink [-t] [-m mapfile] [-n symfile] [-o outfile] [-p pad_value]
               [-s symbol] file [...]
$ rgbfix
usage: rgbfix [-Ccjsv] [-i game_id] [-k licensee_str] [-l licensee_id]
              [-m mbc_type] [-n rom_version] [-p pad_value] [-r ram_size]
              [-t title_str] file
```

https://github.com/bentley/rgbds

BGB Emulator/Debugger





http://bgb.bircd.org/

Everdrive GB

https://krikzz.com/

Game Boy Camera


GAME BOY
camera


GAME BOY
camera


GAME BOY
printer
POWER

http://www.blood-is-red.de


Nintendo
GAME BOY
GAME BOY printer paper
Nintendo


GAME BOY printer
FEED
POWER


Nintendo
Nintendo
Nintendo
Nintendo
Nintendo
GAME BOY
Nintendo
Nintendo
GAME BOY
Nintendo
GAME BOY
Nintendo
GAME BOY
GAME BOY

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Nintendo®

GAME BOY™

Lisa
@ellduin

Dominik
@monkeydom

@_drmr_

Lisa
@ellduin

Dominik
@monkeydom

Costis Sideris

bunnie
@bunniestudios

@_drmr_

Orlando Bassotto

Felix
@tmbinc

The Ultimate Talk Series

25C3 2008

Michael Steil

The Ultimate Commodore 64 Talk

26C3 2009

Sven Oliver ('SvOlli') Moll

The Ultimate Atari 2600 Talk

29C3 2012

Tomaž Šolc

The Ultimate Galaksija Talk

32C3 2015

rahra

The Ultimate Amiga 500 Talk

33C3 2016

Michael Steil

The Ultimate Game Boy Talk

The Ultimate Talk Series

25C3 2008

Michael Steil

The Ultimate Commodore 64 Talk

26C3 2009

Sven Oliver ('SvOlli') Moll

The Ultimate Atari 2600 Talk

29C3 2012

Tomaž Šolc

The Ultimate Galaksija Talk

32C3 2015

rahra

The Ultimate Amiga 500 Talk

33C3 2016

Michael Steil

The Ultimate Game Boy Talk

?

The Ultimate Talk Series

25C3 2008

Michael Steil

The Ultimate Commodore 64 Talk

26C3 2009

Sven Oliver ('SvOlli') Moll

The Ultimate Atari 2600 Talk

29C3 2012

Tomaž Šolc

The Ultimate Galaksija Talk

32C3 2015

rahra

The Ultimate Amiga 500 Talk

33C3 2016

Michael Steil

The Ultimate Game Boy Talk

?

The Ultimate Talk Series

?

The Ultimate Talk Series

?

The Ultimate Talk Series

?

?

I nominate for 34C3

?

?

I nominate for 34C3

I nominate for 34C3

Dominik Wagner

The Ultimate Acorn Archimedes Talk

I nominate for 34C3

Dominik Wagner

The Ultimate Acorn Archimedes Talk

Jannis Harder

The Ultimate Super Nintendo Talk

I nominate for 34C3

Dominik Wagner

The Ultimate Acorn Archimedes Talk

Jannis Harder

The Ultimate Super Nintendo Talk

I nominate for 34C3

Dominik Wagner

The Ultimate Acorn Archimedes Talk

Jannis Harder

The Ultimate Super Nintendo Talk

I nominate for 34C3

Dominik Wagner

The Ultimate
Acorn Archimedes Talk

Jannis Harder

The Ultimate
Super Nintendo Talk

Thank you!