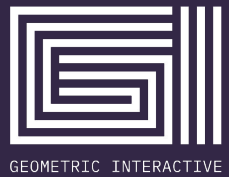
Ancient Game Audio

The Invention of Sound for Video Games

Game Audio Playthrough #46

Jakob Schmid

Geometric Interactive



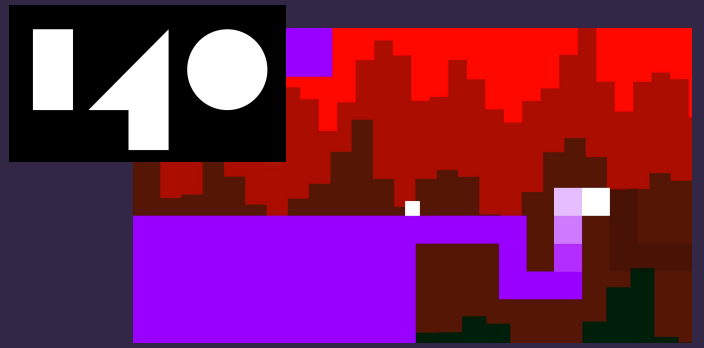
Who am I?

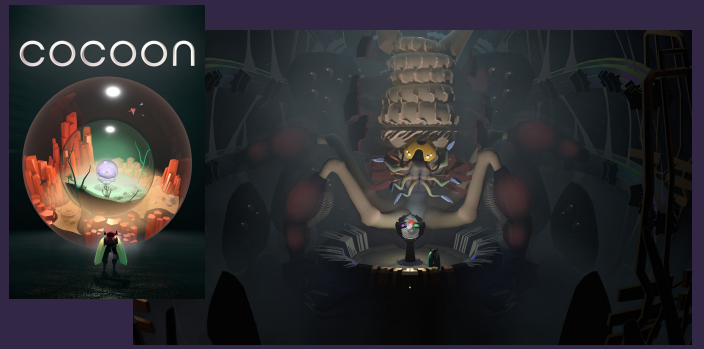
Computer scientist, Aalborg University, Denmark

18 years game development experience

Audio programmer on INSIDE

Geometric Interactive co-founder / audio director





Topics

One Way to Understand Video Game History

Audio Hardware 1970-1980s

One Way to Understand Video Game History

My frame of analysis for this talk:

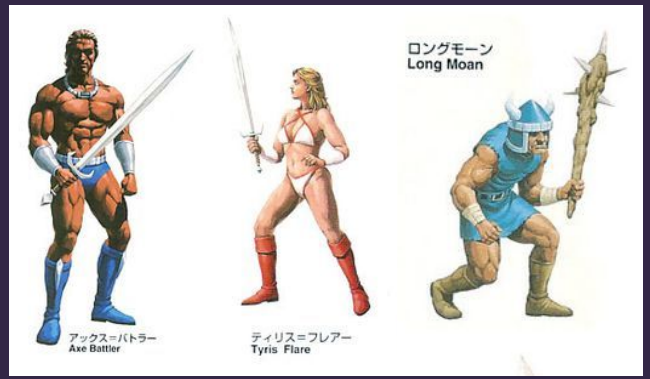
Video games represent a dialogue between artistic ideas and external constraints

Let's look at an example

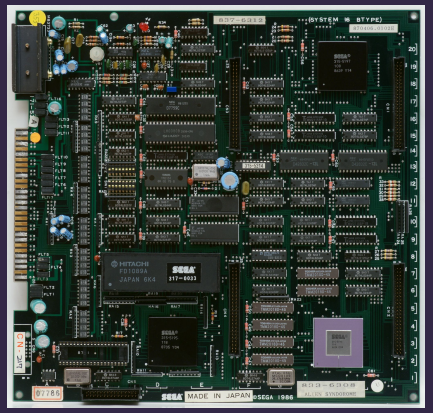
Ideas / Constraints Dialogue Example

Artistic idea:
An action game with fantasy art like Boris Vallejo

Ideas / Constraints Dialogue Example

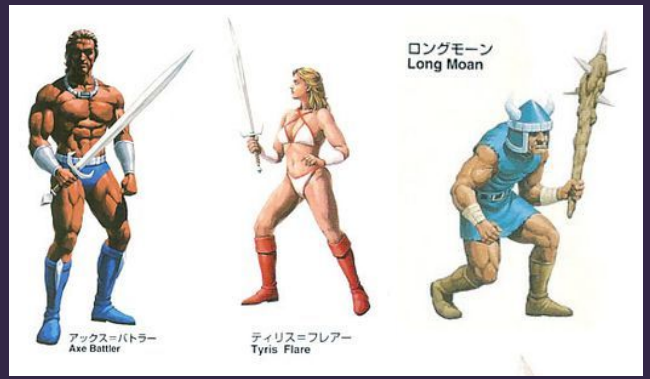


Artistic idea:
Fantasy art like Boris Vallejo

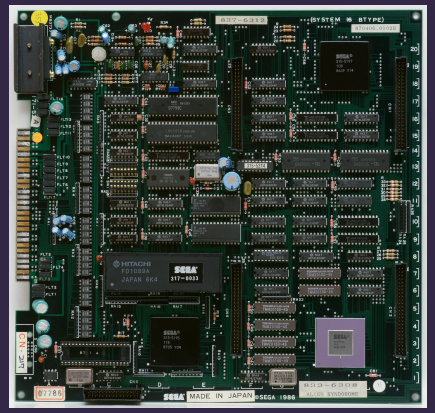


Hardware constraints:
Sega System 16 (1985)

Ideas / Constraints Dialogue Example



Artistic idea:
Fantasy art like Boris Vallejo



Hardware constraints:
Sega System 16 (1985)

End result

```
Title     : Golden Axe
Platform  : Arcade (System 16)
Year      : 1989
Developer : Sega
```

How can we observe this dialogue?

Noticing Hardware Constraints

```
Title     : Zombies Ate my Neighbors
Platform  : Sega Mega Drive
Year      : 1993
Developer : Lucasfilm Games
```

What graphical constraints can we see here?

Noticing Hardware Constraints

Background graphics are constrained to 8x8 grid!

Noticing Hardware Constraints

Title : Congo Bongo
Platform : ColecoVision
Year : 1984
Developer : Sega

Title : Girl's Garden
Platform : SG-1000
Year : 1985
Developer : Sega

Title : Penguin Adven
Platform : MSX
Year : 1986
Developer : Konami

What do these games have in common?

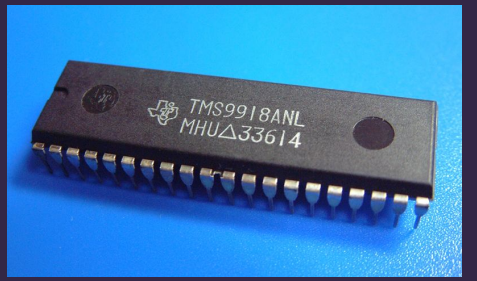
Noticing Hardware Constraints

256x192 resolution with 15-color fixed palette
32 single-color sprites

Why do 3 games for 3 different platforms have exactly the same graphics constraints?

TMS9918

MSX, ColecoVision, Sega SG-1000
Same graphics chip: TMS9918




```
Platform  : MSX
Year      : 1983
Developer : Microsoft/ASCII Corp
```

```
Platform  : ColecoVision
Year      : 1982
Developer : Coleco Industries
```


```
Platform  : SG-1000
Year      : 1983
Developer : Sega
```

Ideas / Constraints Dialogue

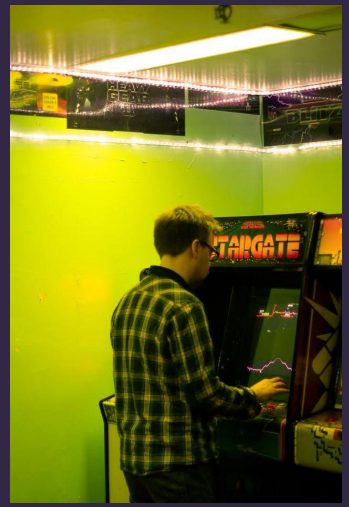
- Hardware constraints affect the execution of ideas
- Same for development constraints
- Sometimes they inhibit, sometimes they inspire

Disclaimer

- I don't see constraints as a negative
- I impose artificial constraints on myself (and others) as inspiration
- Many of the developers, sound designers, composers we will talk about today agree

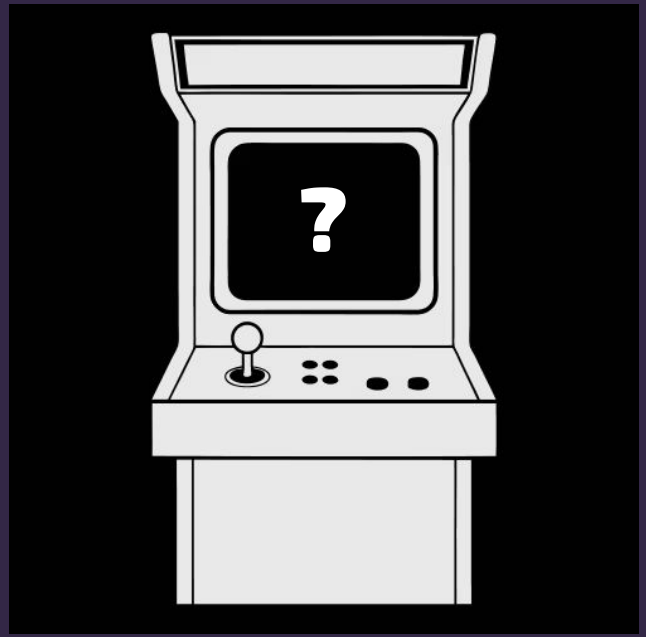
Video Games 1970-1980

- Arcade games was the cutting edge of the medium
- Sound was essential



Thought Experiment

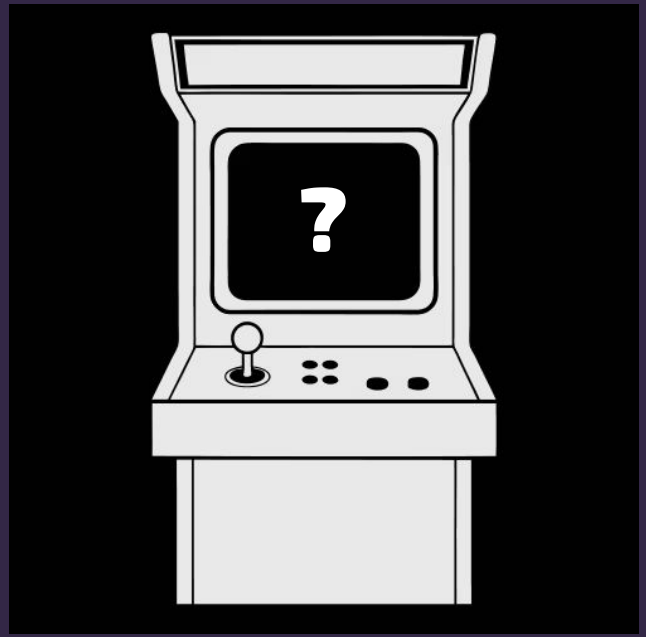
It's 1980 and you're designing an arcade machine



Thought Experiment

It's 1980 and you're designing an arcade machine.

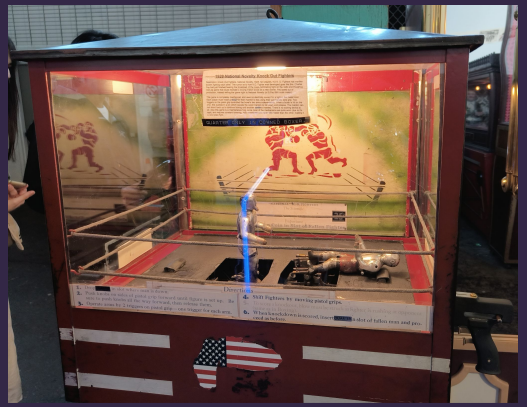
So, what hardware are you using for sound?



Mechanical Arcade Games

Existed since early 1900s
Pinball machines and other electro-mechanical arcade games


```
Title     : Ace High
Year      : 1957
Platform  : Pinball
Developer : Gottlieb
```



```
Title     : Knock Out Fighters
Year      : 1928
Platform  : Arcade machine
Developer : National Novelty
```


```
Title     : Sky Raider
Year      : 1958
Platform  : Arcade machine
Developer : United
```

Mechanical Sound: Pinball

Physical sounds
- Ball
- Flippers
- Bumpers

Electro-mechanical sounds
- Clicks
- Bells


```
Title   : Old Chicago
Year    : 1976
Company : Bally
Sound   : Electro-mechanical
```

▸ gap26-old_chicago

How does that work?

Electro-Mechanical Pinball Sounds

How do the electro-mechanical sounds work?

Let's ask a YouTube guy

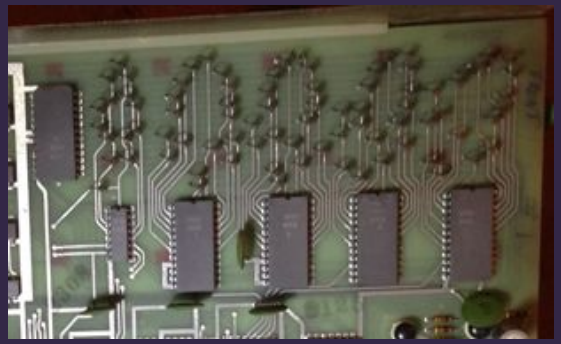

```
Title   : Aztec
Year    : 1976
Company : Williams
Sound   : Electro-mechanical
```

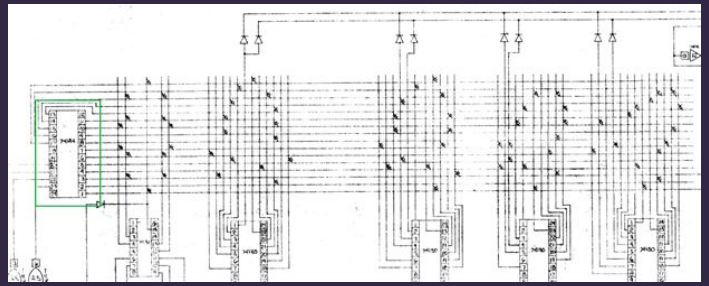
First Arcade Video Game

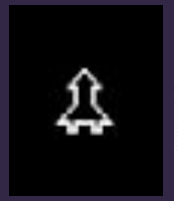

```
Title   : Computer Space
Year    : 1971
Company : Syzygy Engineerings
```

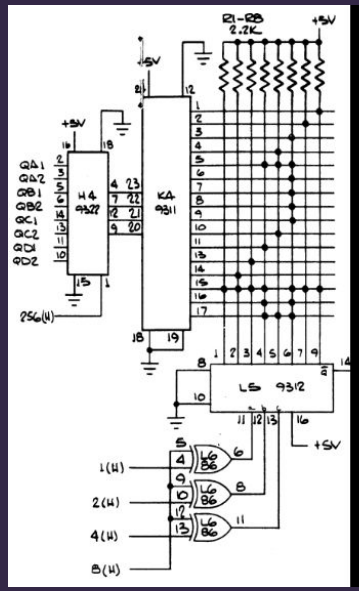
How were games designed in the early 1970s?

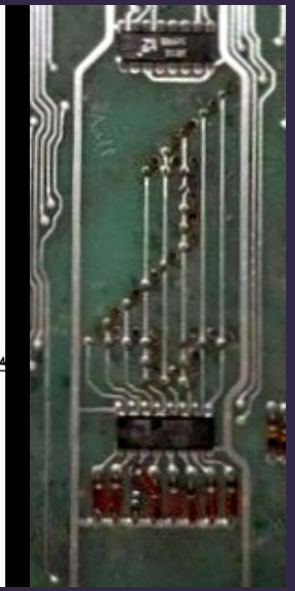
No CPU, No Software, No ROM Data











```
Title   : Computer Space
Year    : 1971
Company : Syzygy Engineerings
```

```
Title   : Space Race
Year    : 1973
Company : Atari
```

Let's look at a game in detail!

Pong

- Two-player tennis game
- Controlled with analog knobs
- Assigned as an exercise for new hire Allan Alcorn
- Designed in 2 months
- First commercially successful video game, sold 35,000 units
- Profit 17.5 m$ (~ 135 m$ in 2026) on arcade units alone


```
Title     : Pong
Year      : 1972
Platform  : Arcade machine
Developer : Atari
```

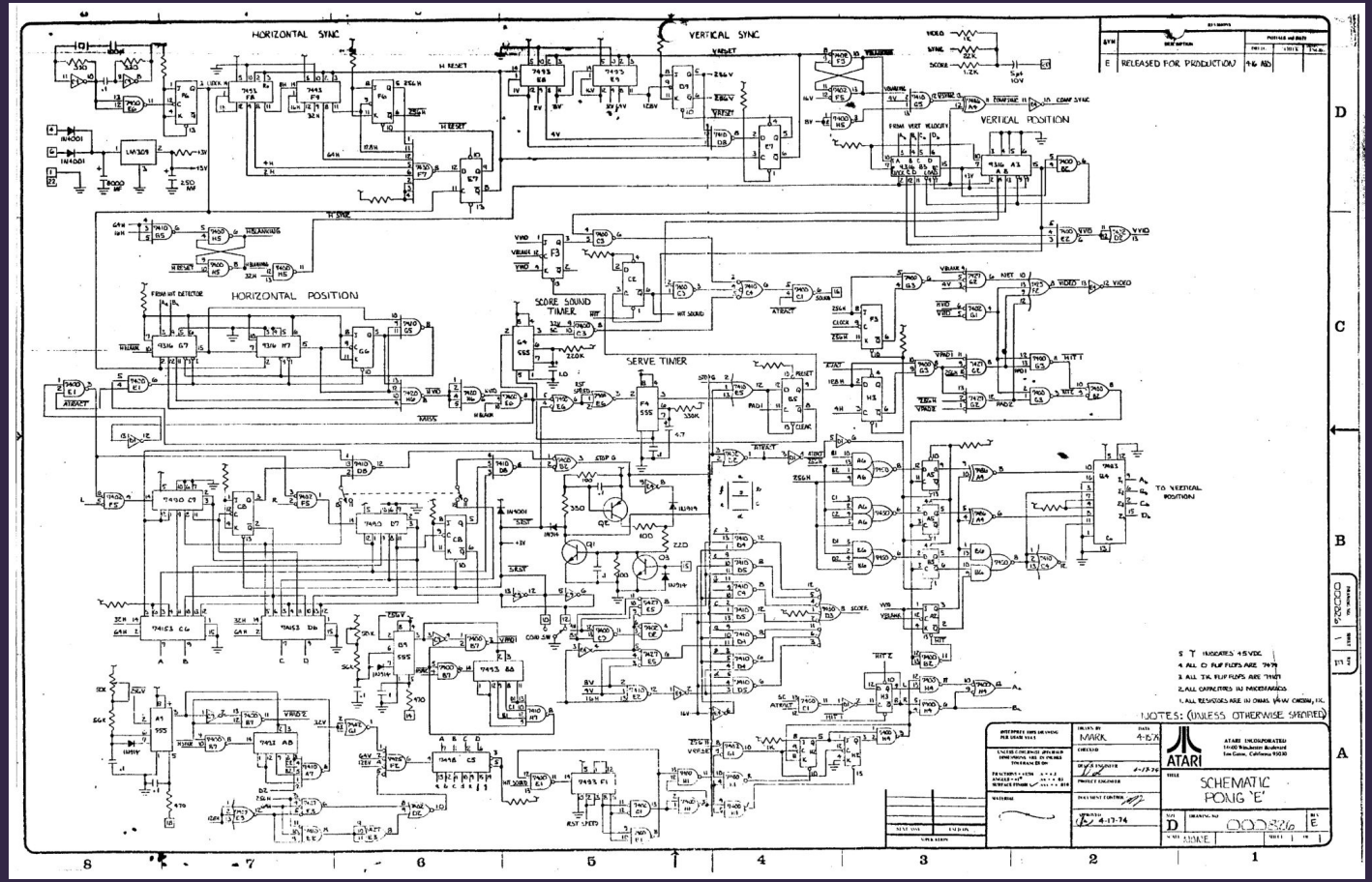
How was it designed?

Pong

This is Pong:

- Game logic
- Art
- Sound
- Video controller

Everything!



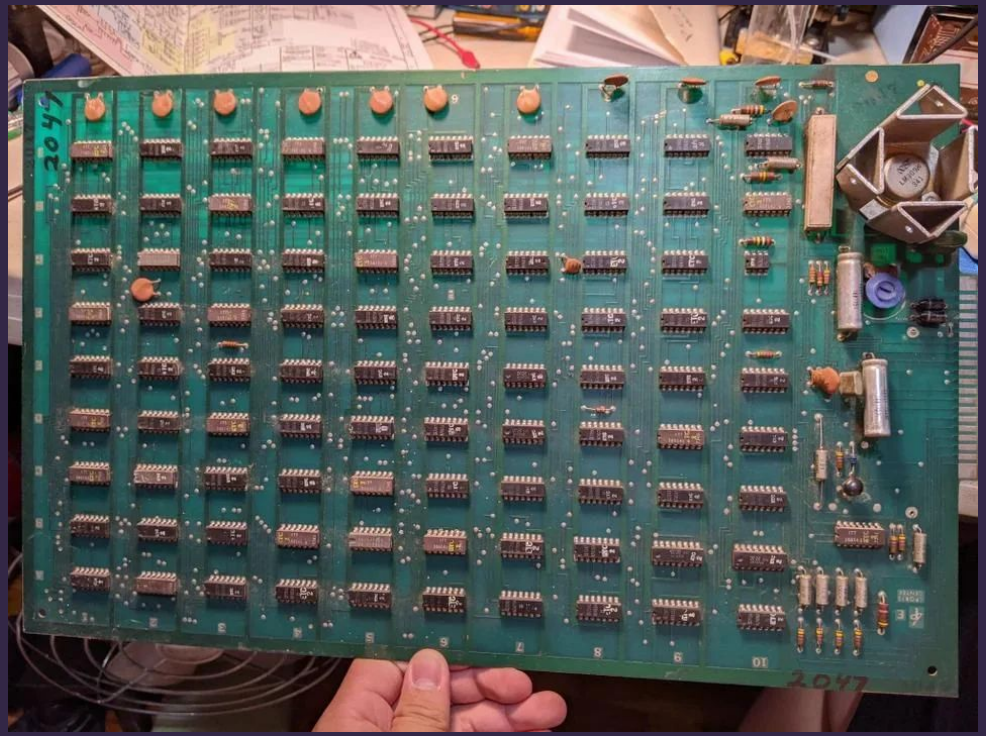
What are all these boxes with numbers?

Transistor-Transistor Logic Chips

Small chips with <14 pins
Manufactured from 1963 forward
Simple binary operations

```
--------------------------------------------------
List of Pong components

TTL
Part #   Function
--------------------------------------------------
7400     NAND
7410     3-input NAND
7430     8-input NAND
7474     2xflip flop (2 x 1-bit memory)
74107    2xflip flop with clear
555      2.1 MHz timer
7493     4-bit binary counter (divide by 2, 4, 8)
9316     4-bit up/down binary counter
--------------------------------------------------
```



Does Pong have sound?

Pong Sound

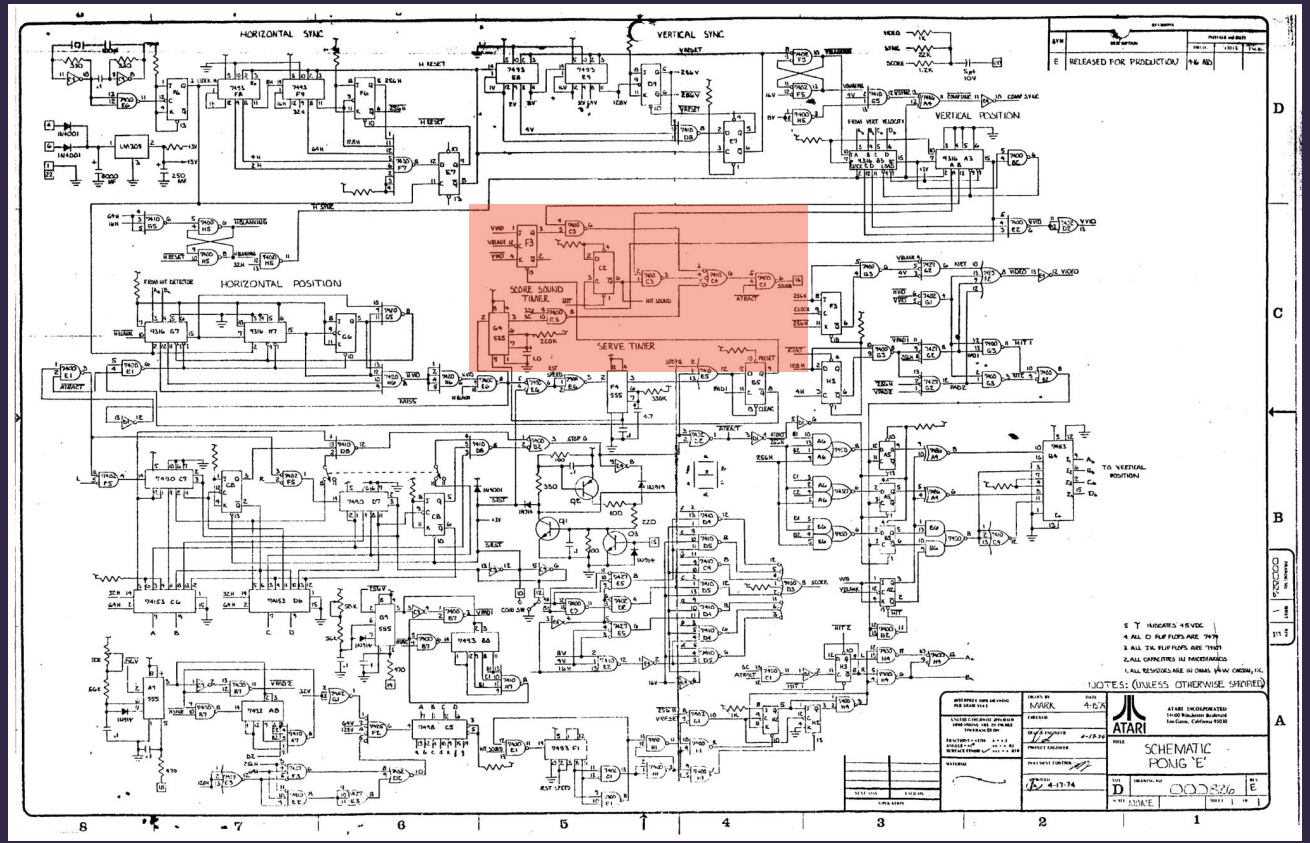
Pong has 3 sounds:

TOP BOTTOM HIT	(short 246 Hz square wave)
PADDLE HIT	(short 592 Hz square wave)
SCORE	(longer 246 Hz square-ish wave)


```
Title     : Pong
Year      : 1972
Platform  : Arcade machine
Developer : Atari
```

How do I know those frequencies so accurately?

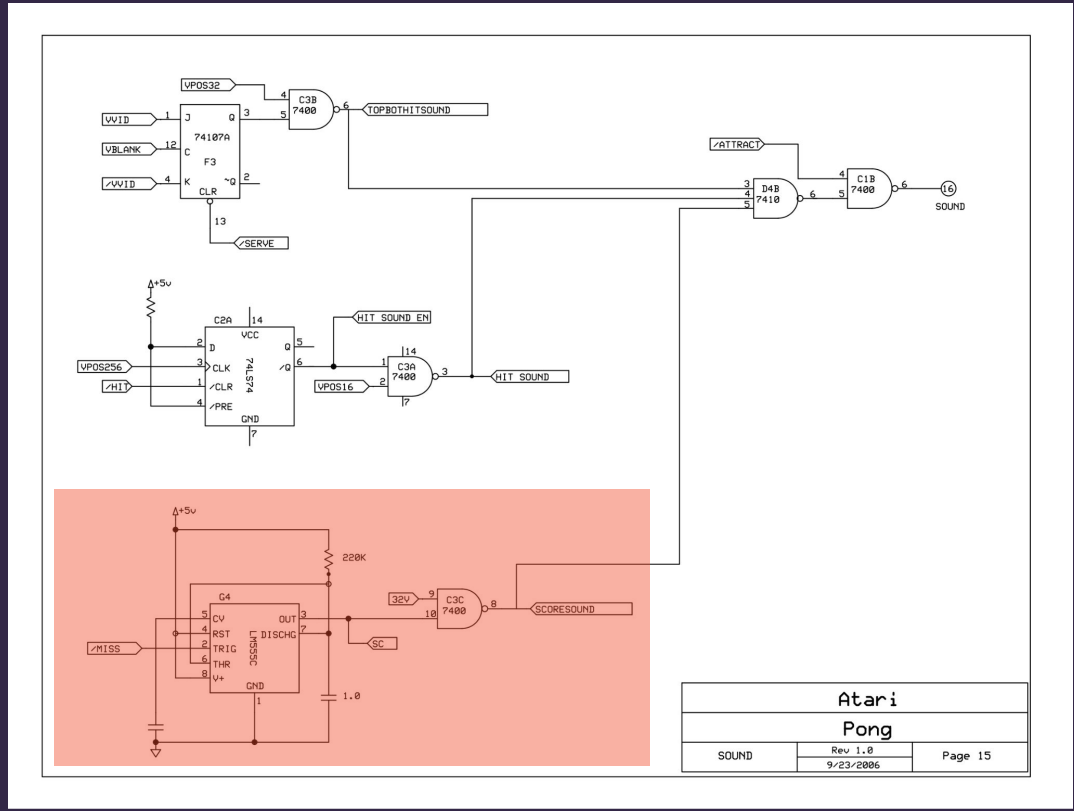
Pong Sound



How does it work?

Sound Circuits

- Each sound has its own circuit
- Marked is the SCORE sound circuit



Let's look closer the SCORE sound

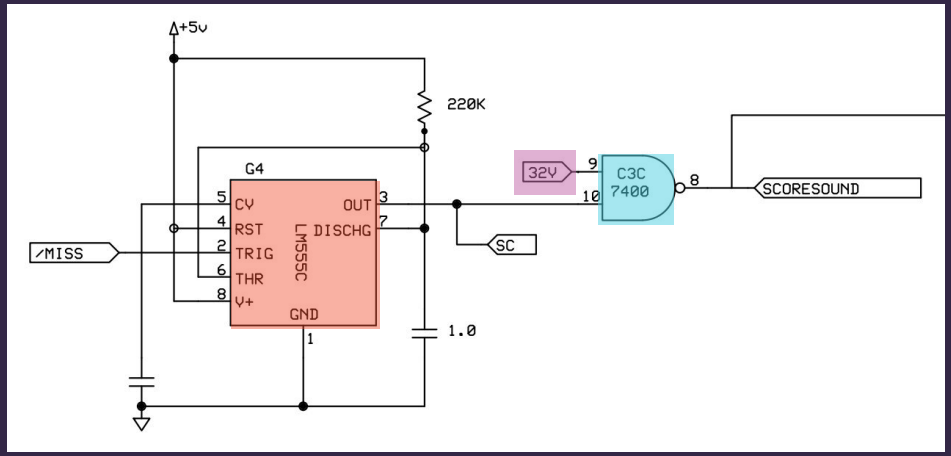
SCORE Sound

Components:

555 : timer

7400 : NAND (NOT AND operation - works here as a binary AND)

- 555 triggered by /MISS when ball goes out of bounds
- OUT goes high for 242 ms
- Gates the 32V signal to output

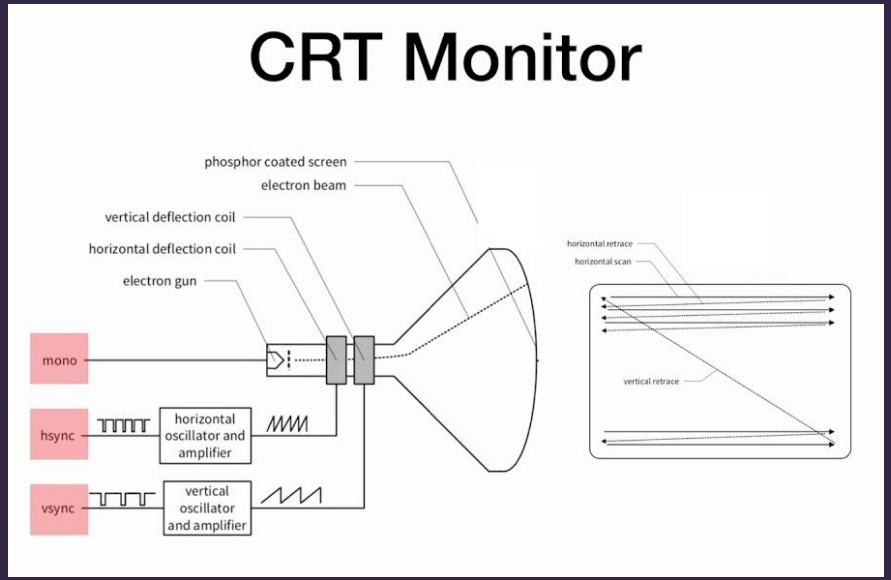


What is the 32V signal?

Master Clock and Video Synchronization

Pong is a computer-like device that directly controls a CRT display.

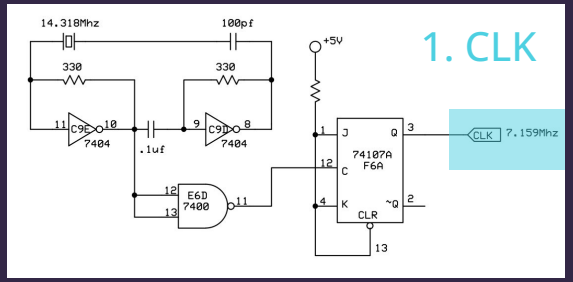
- The whole board runs using a master clock (CLK)
- CLK is used to generate video synchronization signals (VSYNC/HSYNC)
- Outputs luminance (B&W) video signal
- And mono sound

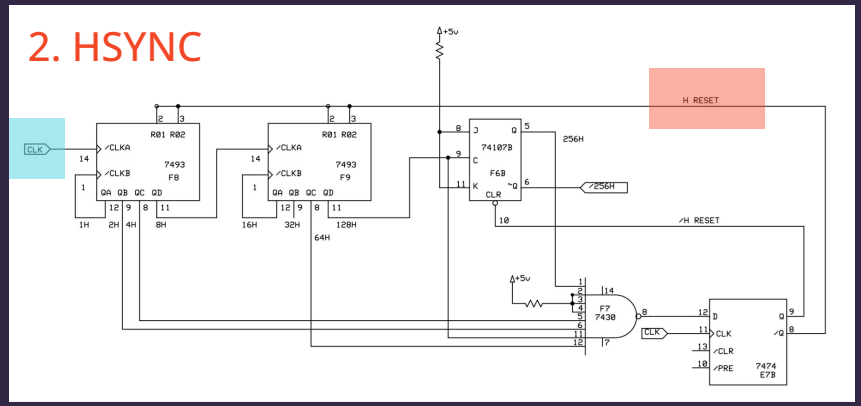


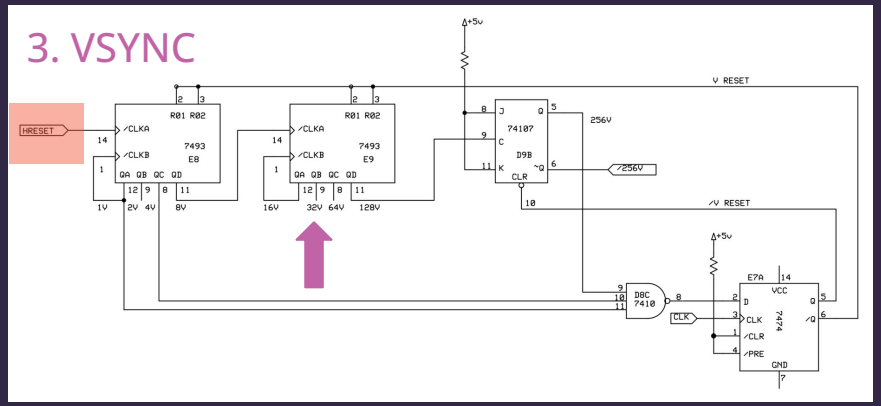
How's that related to the 32V signal?

From Crystal to 32V

1. 14.3 MHz crystal generates 7.16 MHz master clock CLK
2. HSYNC counts 455 CLKs, outputs HRESET at 15.7 KHz
3. VSYNC counts 262 CLKs, outputs VRESET at 60.0 Hz
4. During VSYNC counting, 6th binary digit is 32V, a 246 Hz square wave (glitchy, because it resets every 1/60 s)





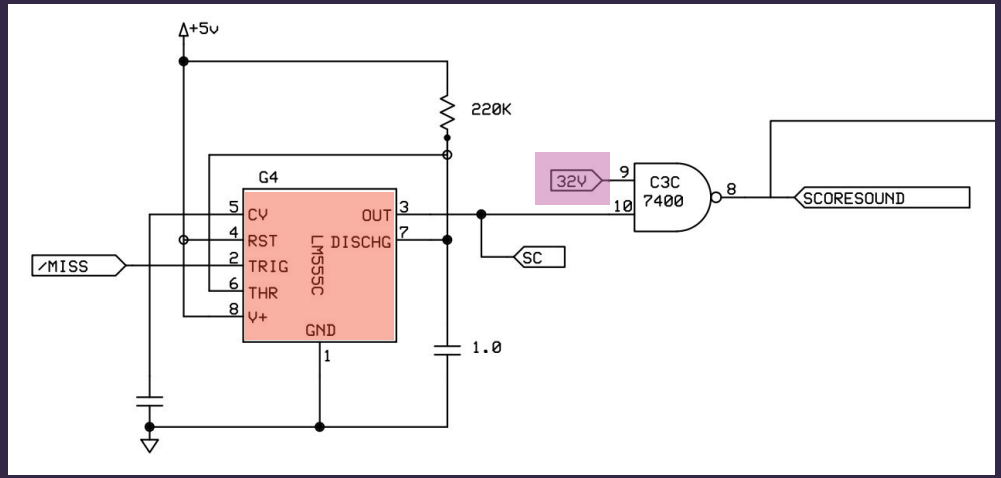


SCORE Sound

Now we know where the sound comes from:

- 32V is a 246 Hz digital square wave
- 555 acts as a gate for 242 ms

The output is a 1-bit digital signal

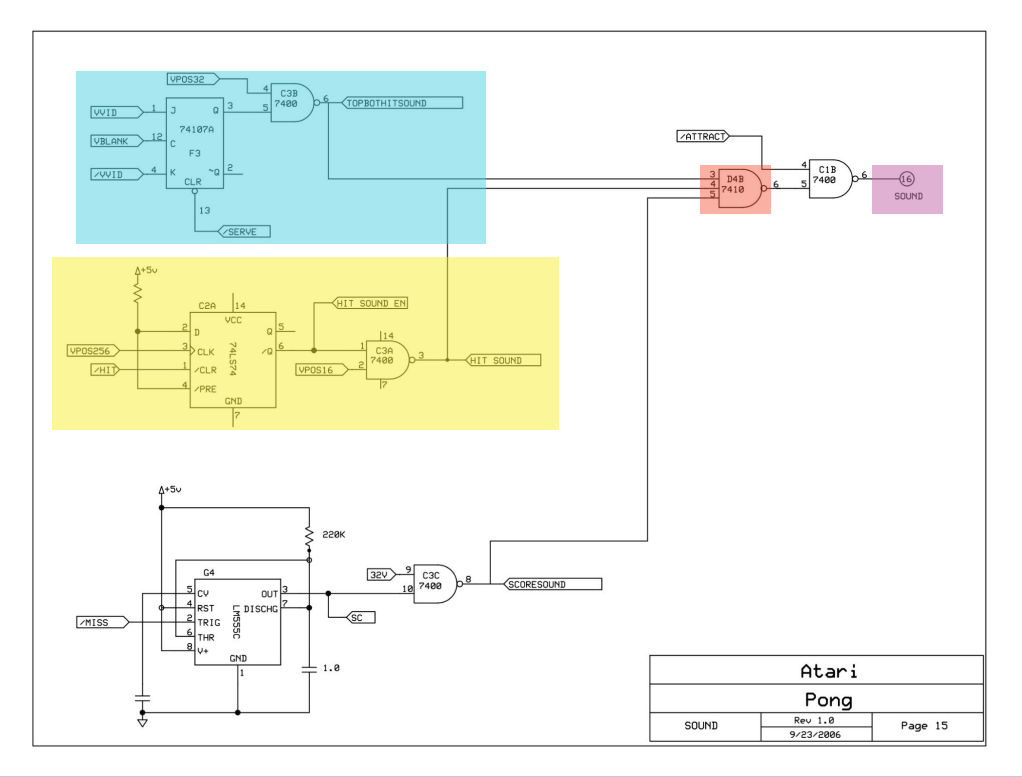


Sound Circuits

SCORE sound gets digitally mixed with two other sounds using NAND gate:

TOP BOTTOM HIT
PADDLE HIT

And sent to the SOUND output



Digital Hardware Synthesis

- No CPU, no software, no ROM data
- Designed like a computer
- Implemented using discrete TTL logic
- 1-bit digital audio, 3 different sounds
- Generated from video sync clocks

How did the designer approach the audio?

Idea / Constraints

Al Alcorn, designer of Pong:

I was running out of parts on the board.

Nolan wanted [..] the approving roar of cheering people when you made a point.
Ted Dabney told me to make a boo and a hiss when you lost a point [..].

I said, “Screw it, I don’t know how to make any one of those sounds. I don’t have enough parts anyhow.”
Since I had the wire wrapped on the scope, I poked around the sync generator to find an appropriate frequency or a tone.

So those sounds were done in half a day. They were the sounds that were already in the machine.

Analog Hardware Synthesis

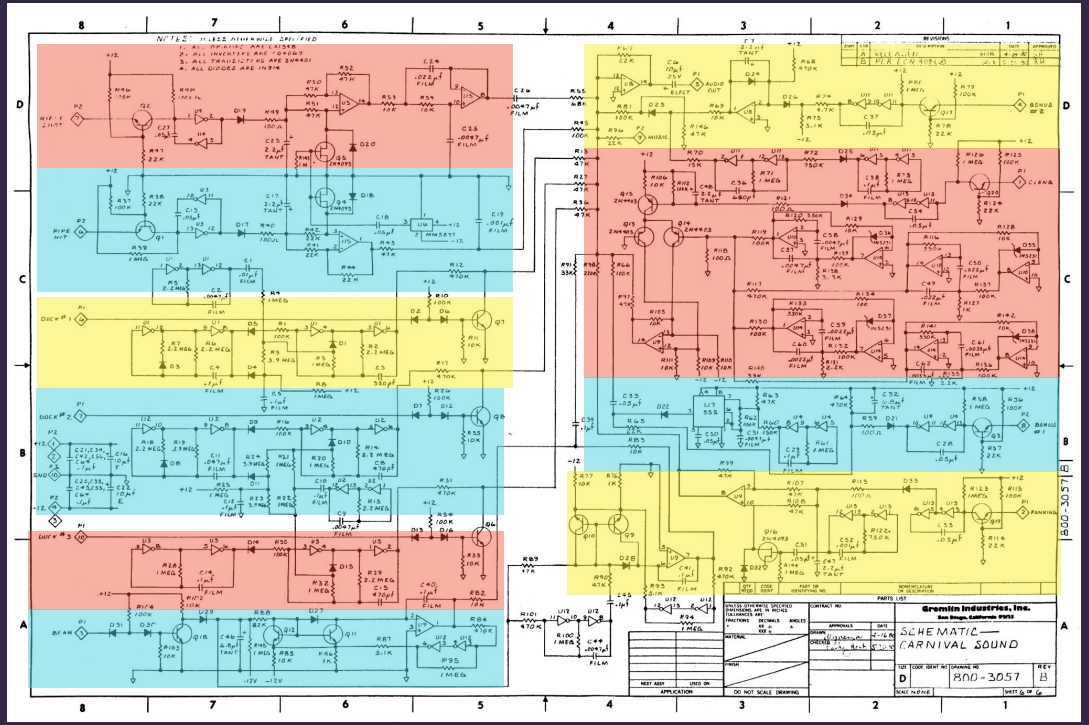
- Carnival by Sega (1980)
- Gallery shooting game
- Capture experience of going to a traveling carnival
- Runs on a Z80 CPU
- In-game music toggle!


```
Title     : Carnival
Year      : 1980
Platform  : Arcade machine
Developer : Sega
Sound     : Analog H/W Synthesis
```

▸ gap26-carnival

Carnival Sound Board

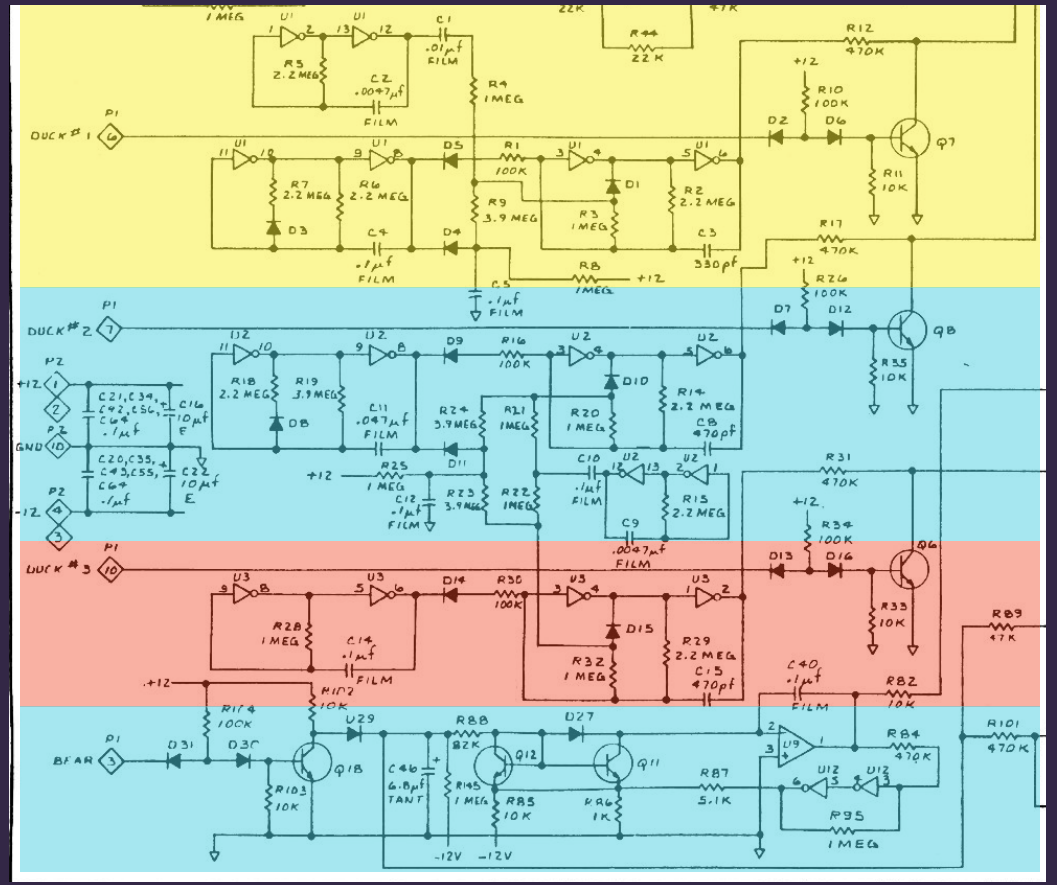
- Sound only, music is its own board
- Individual sounds triggered on left and right sides



Let's look closer!

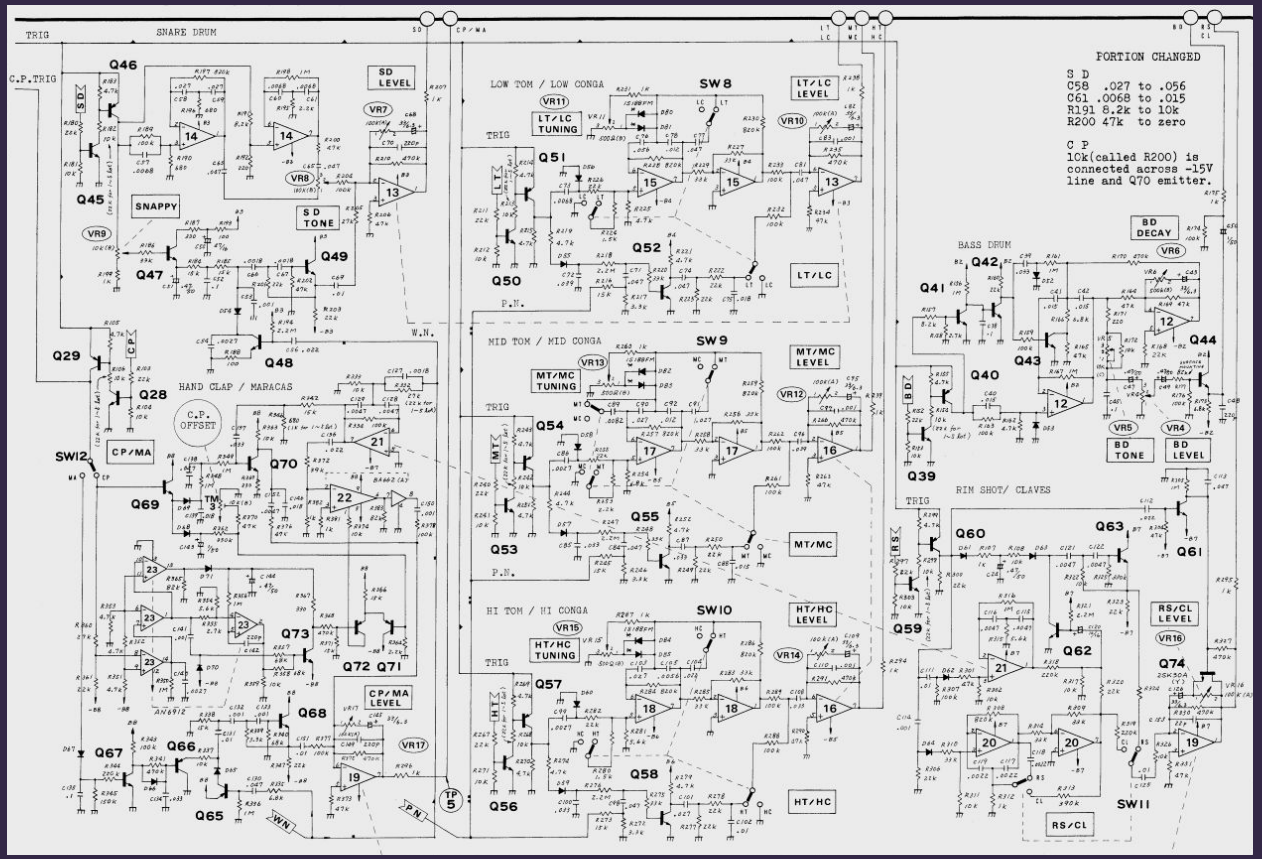
Carnival Sound Board

- Each sound has its own analog circuit
- Using amps, inverters, transistors, diodes
- Here we see sounds DUCK #1-3 and BEAR



This reminds me of something...

Analog Hardware Synthesis



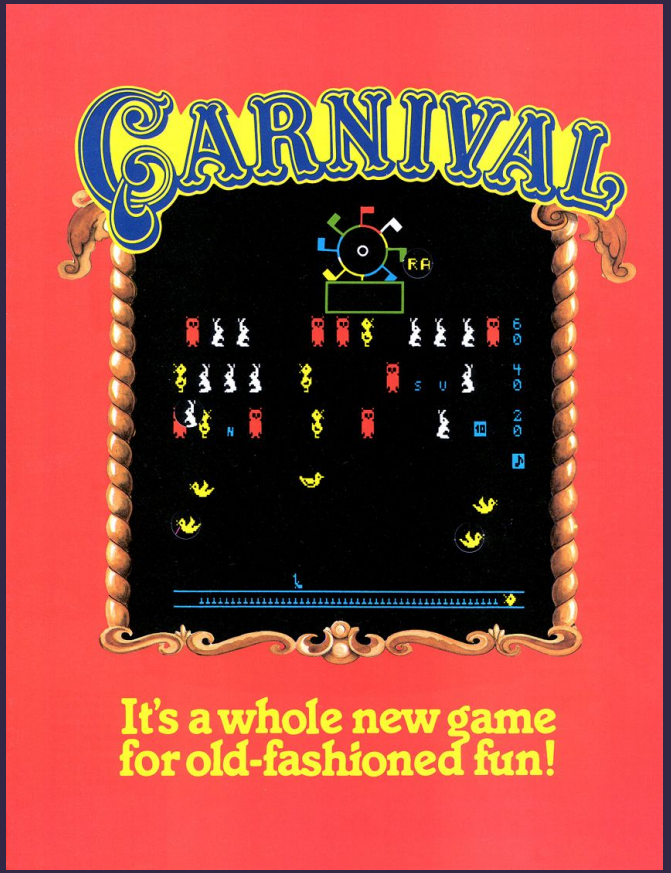
What is this?

Idea / Constraints

I couldn't find information about the development of Carnival.

But we can speculate:

- Unique hardware was designed for every sound effect. The designer of Carnival seemingly wanted players to have a realistic shooting gallery experience
- Music was added to enhance carnival ambience



OK, and how does the music work?

Carnival Music

Uses General Instruments AY-3-8910 (1978) - more on this chip later

Is this the first video game music?

Inventing Music

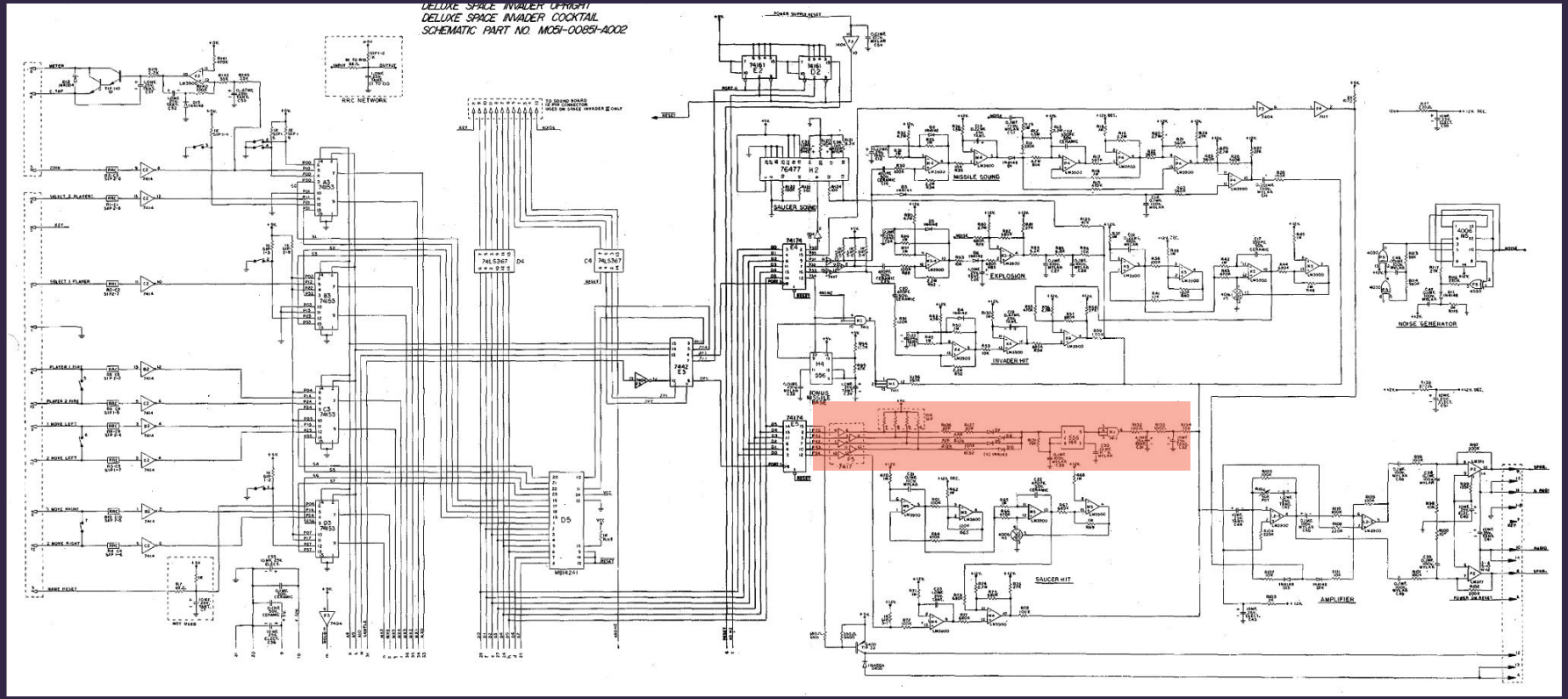
- Space Invaders!
- Immensely succesful worldwide
- Generated 4 b$ in 1978-1982 (10 b$ adjusted for inflation)
- Analog sound board similar to Carnival


```
Title     : Space Invaders
Year      : 1978
Platform  : Arcade machine
Developer : Taito
Sound     : Analog H/W Synthesis
```

▸ gap26-space_invaders

How does the music work?

Space Invaders Music



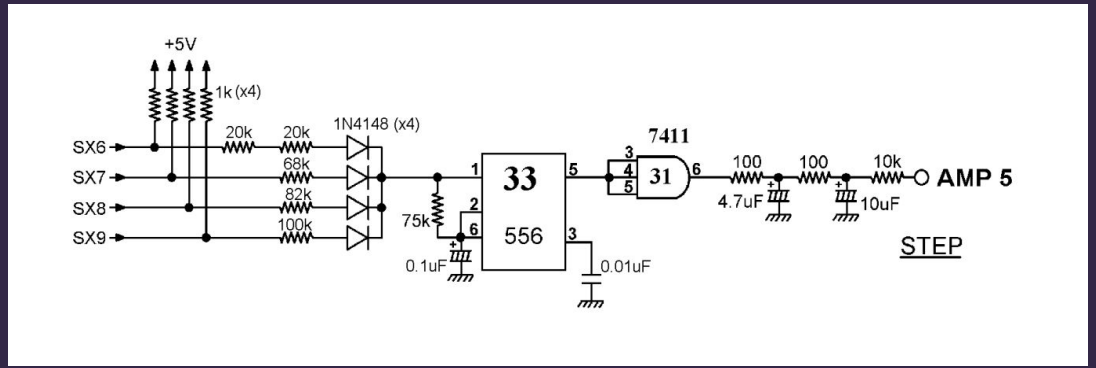
How's that music?

Space Invaders Music

- 556 is a timer (like 555 in Pong)
- Works as an oscillator in this configuration
- Generates pitch based on the input resistance
- 4 different resistor combinations are selected

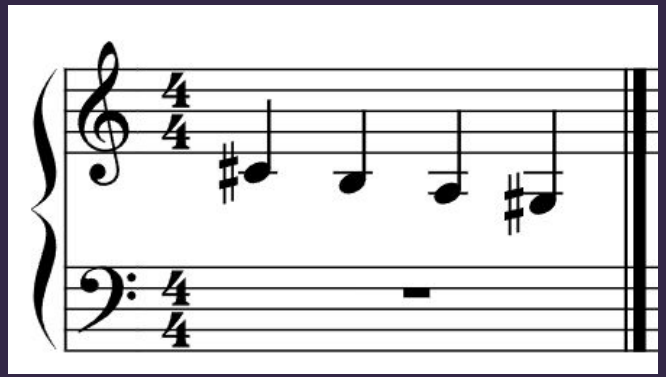
Generating frequencies:

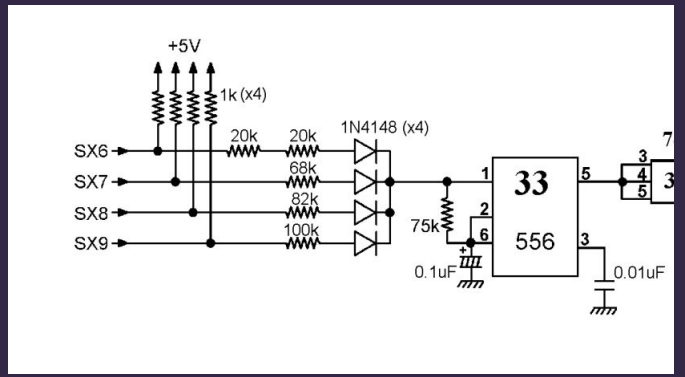
34.9 Hz
30.0 Hz
28.1 Hz
25.8 Hz



Space Invaders Music Notes

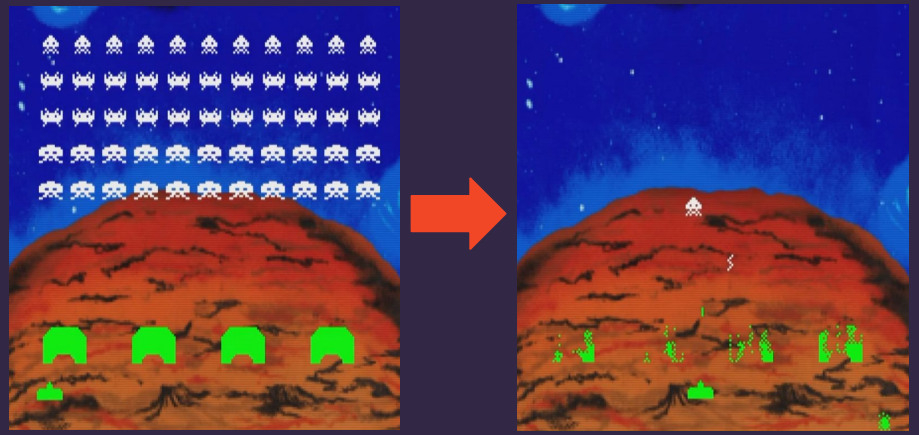
34.9 Hz ~ C#1
30.0 Hz ~ B-0
28.1 Hz ~ A-0
25.8 Hz ~ G#0





Adaptive Tempo

- Tempo increases as gameplay gets faster
- Starts at 69 BPM and accelerates to 690 BPM (when shooting the final alien)



69 BPM

690 BPM

Idea / Constraints

Taito audio engineer Michiyuki Kamei:

- Worked on 'Blue Shark' and 'Space Invaders' at the same time, around 4-5 months.
- Modified 'Blue Shark' sound board for 'Space Invaders' due to time pressure
- Shot sound has most components, because of its importance
- Music originally played at higher pitch, but was lowered to fit scary cabinet art. Inspired by John Williams' 'Jaws' theme (1975).
- Cabinet speakers broke from too much bass. Taito employees had to go and replace speakers with stronger ones.

Was this the first music in video games?

The Amazing Maze Game

Listen to this and observe the gameplay


```
Title     : The Amazing Maze Game
Year      : 1976
Platform  : Arcade machine
Developer : Midway
Sound     : Analog H/W Synthesis
```

▸ gap26-amazing_maze

Question Time coming up!!

Question Time!

Two questions:

1. How were the tones selected?

Question Time!

Two questions:

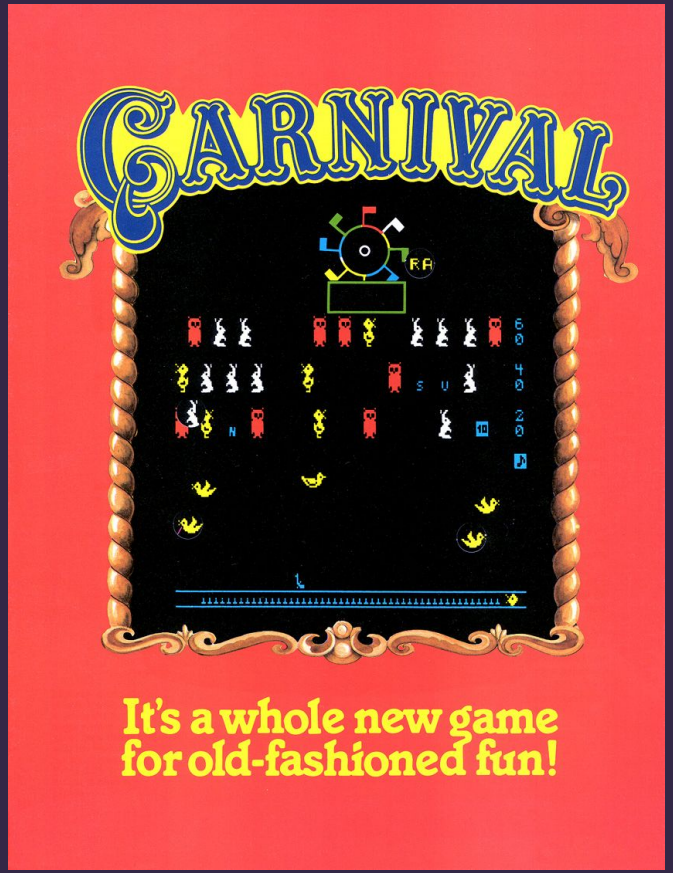
1. How were the tones selected?
2. Is this music?

If yes, this is possibly the first video game with music.

Let's go back to the Carnival music...

Carnival Music

- General Instruments AY-3-8910 (1978)
- This is a Programmable Sound Generator (PSG)



Are there more PSGs?

Programmable Sound Generators

- Fixed-function sound generation chips
- Digital hardware synthesis in a convenient chip


```
Title     : AY-3-8910
Year      : 1978
Developer : General Instruments
Type      : 3 x Square + noise
Used in   : Arcade games
            Intellivision (1979)
            Vectrex (1982)
            MSX (1983)
            Atari ST (1985)
            ZX Spectrum 128 (1985)
```


```
Title     : 6581
Year      : 1982
Developer : MOS Technology
Type      : Digital oscillators
            Analog filter
Used in   : Commodore 64
```

Let's check out the AY...

AY-3-8910


```
Platform  : MSX
Year      : 1983
Developer : Microsoft/ASCII Corp
Sound     : AY-3-8910
```

```
Title     : Xenon
Year      : 1980
Platform  : Pinball
Developer : Bally
Sound     : Samples, AY-3-8910,
            mechanical
```

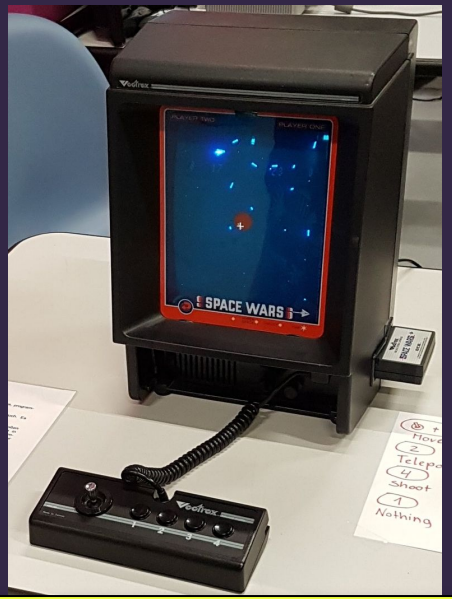

```
Title     : Carnival
Year      : 1980
Platform  : Arcade machine
Developer : Sega
Sound     : 2 x AY-3-8910
```

Any other uses?

AY-3-8910

Platform : Atari ST
Year : 1985
Developer : Atari
Sound : AY-3-8910

Platform : Intellivision
Year : 1979
Developer : Mattel
Sound : AY-3-8910

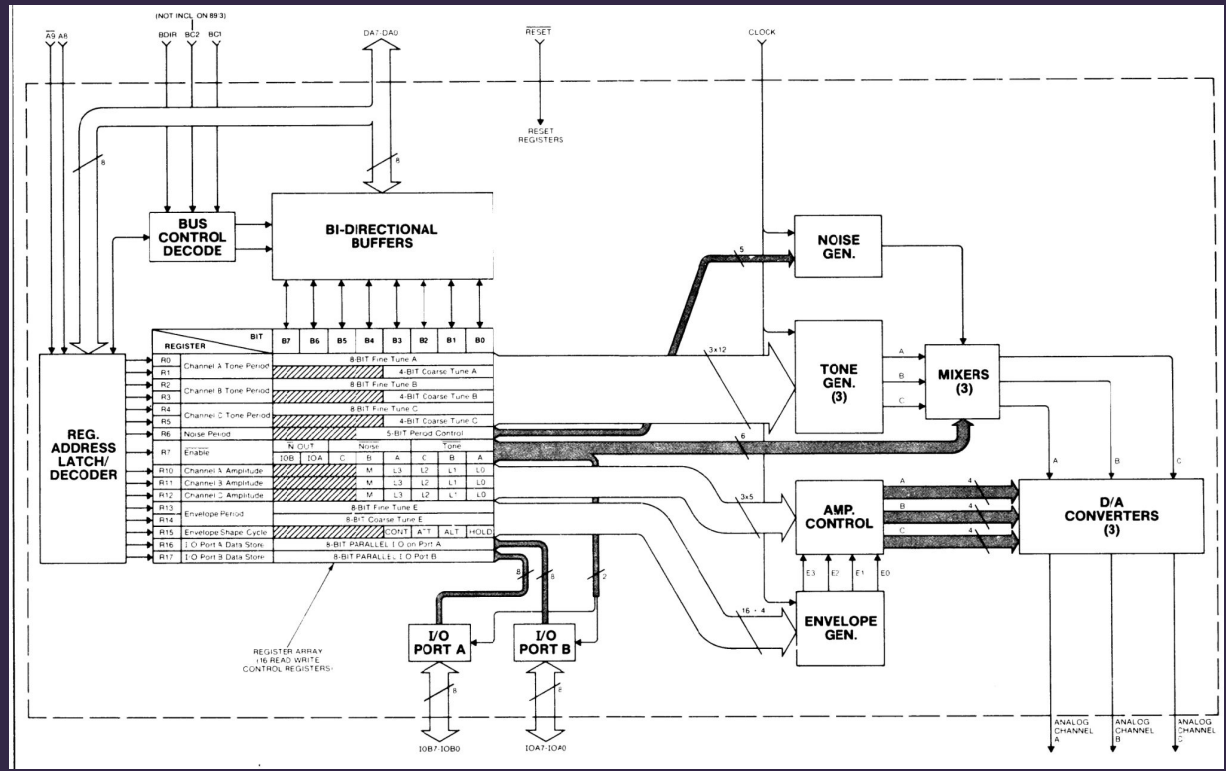


Platform : Vectrex
Year : 1979
Developer : General Consumer Electronics
Sound : AY-3-8910

How does it work?

AY-3-8910

- 3 square wave oscillators
- 12-bit frequency
- Amplitude control
- Fixed envelope shapes (can be envelopes or LFOs)
- Noise - 5-bit freq. control
- Fully digital, converts with built-in D/A converter



L.E.D. Storm

Music by Tim Follin


```
Title     : L.E.D. Storm
Year      : 1988
Platform  : ZX Spectrum 128
Developer : Software Creations
Sound     : AY-3-8910
```

▸ gap26-led_storm

What about the SID chip?

SID

The Commodore 64 is the best selling home computer in history (>12 million units)

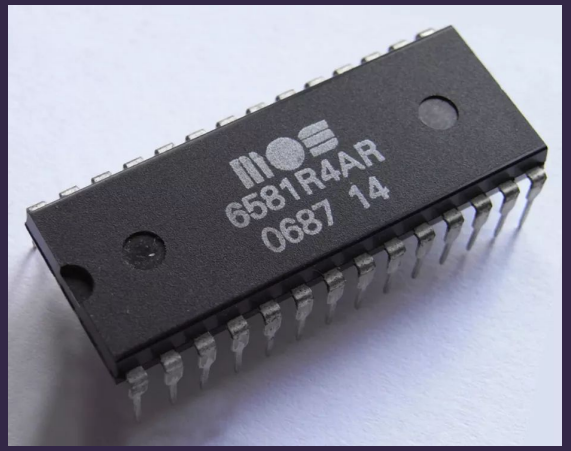
SID chip designed by Bob Yannes, who later co-founded Ensoniq

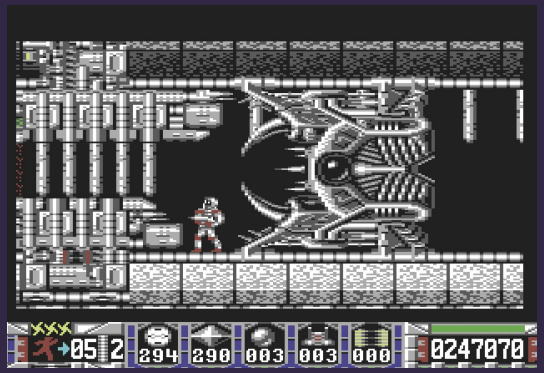

```
Title     : 6581
Year      : 1982
Developer : MOS Technology
Type      : Digital oscillators
            Analog filter
Used in   : Commodore 64
```

How does it work?

Commodore 64 SID Chip

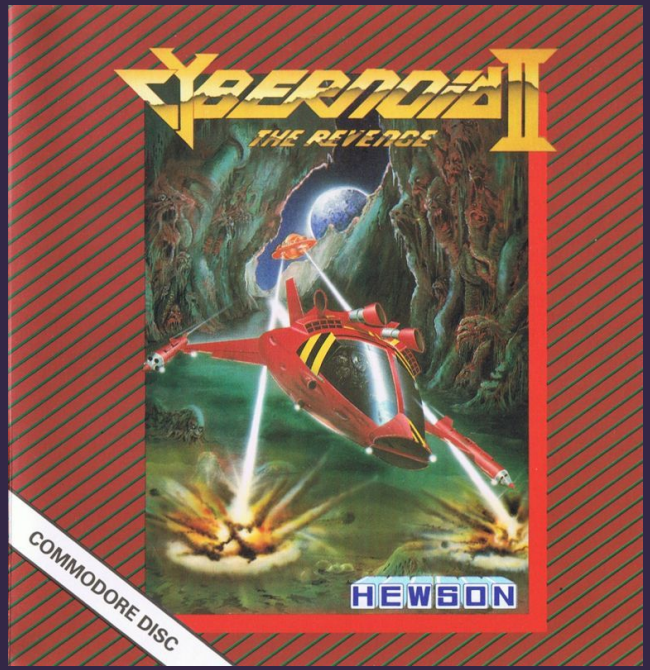
- 3 digitally controlled oscillators
- Selectable waveforms: pulse, triangle, saw, noise
- Can change waveform on the fly
- Ring modulation, oscillator sync
- Multimode filter: low-, high-, bandpass (6dB/12dB rolloff)
- 3 Envelope generators





Cybernoid II: The Revenge

Music by Jeroen Tel




```
Title     : Cybernoid II: The Revenge
Year      : 1988
Platform  : Commodore 64
Developer : Hewson Consultants
Sound     : SID
```

▸ gap26-cybernoid2

What else can the SID chip do?

Inventing Ambience

- Ambience was rarely used in arcade games, possibly because arcades tend to have their own loud ambience already.
- More common in home computer games
- More common in European games than US/Japanese games
- SID chip was capable of synthetic ambience


```
Title     : Arctic Shipwreck
Year      : 1983
Platform  : Commodore 64
Developer : F451 Software
Sound     : SID
```

▸ gap26-c64-ambience

What were the C64 developers going for?

Idea / Constraints

We can speculate about developer ideas:

- Ambiences simulating real world phenomena
- Waves, wind, thunder, crowd noise, room tones

Harsh constraints:

- SID chip has 3 oscillators and a filter for all sound
- Games need sound effects, so either ambience or music!

Expert composers wrote around the 3 voice limitation to make amazing music.

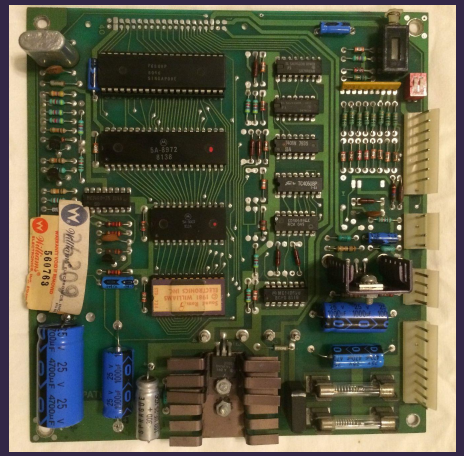
Software Synthesis

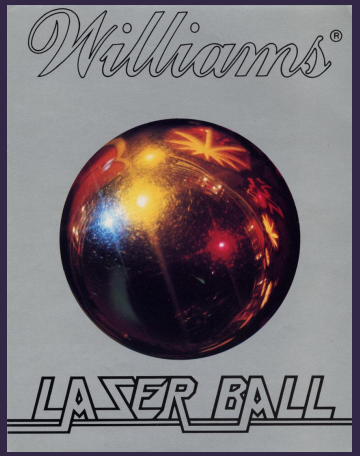
- The availability of CPUs and ROM chips enabled software synthesis
- In the mid-1970s, arcade manufacturers started using CPUs

Can the CPU be used to generate sound?

GWave

- Software synthesis sound board
- Eugene Jarvis created for 'Laser Ball' (1979)
- MC6800 CPU running code from ROM
- Renders 8-bit sound output

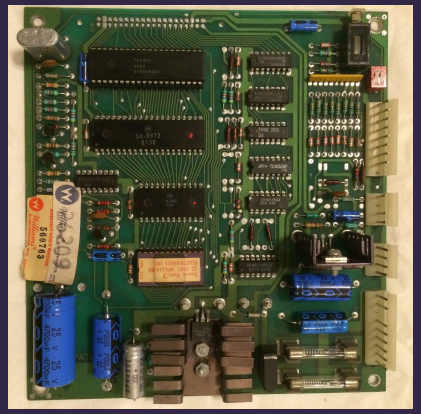





```
Title   : Laser Ball
Year    : 1979
Company : Williams
Platform: Pinball
Sound   : Software synthesis
```

How powerful is that CPU?

Laser Ball vs. Fairlight CMI



Laser Ball Sound Board (1979)

CPU: 1 x MC6800
2 KB ROM
128 B RAM

Output: Variable freq. 8-bit samples

Fairlight Computer Musical Instrument (1979)

CPU: 2 x MC6800
8 * 16 KB sample RAM
64 KB system RAM

Output: 24 KHz 8-bit samples

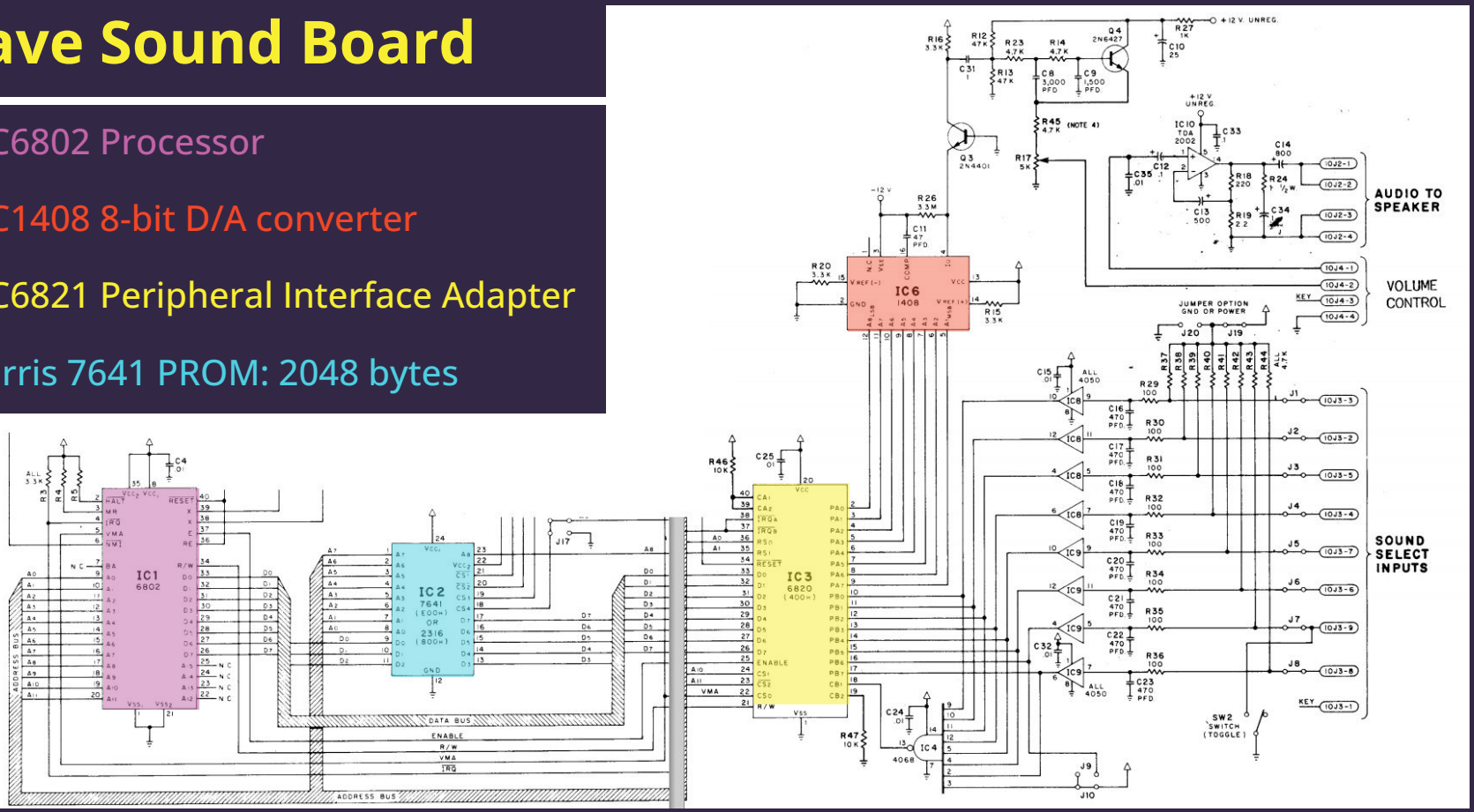
GWave Sound Board

MC6802 Processor

MC1408 8-bit D/A converter

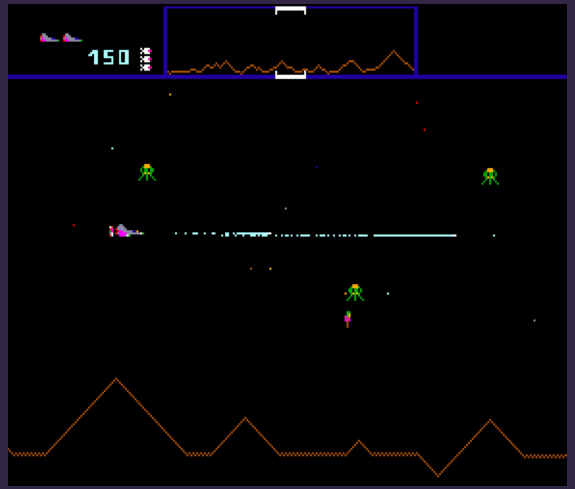
MC6821 Peripheral Interface Adapter

Harris 7641 PROM: 2048 bytes



Defender

- Super-intense scrolling shooter from 1981
- Reused Laser Ball GWave sound board
- ROM was updated, new sounds



Defender Sound ROM

2048 bytes for all Defender audio

Defender Sound ROM, 2048 bytes

```
FF 0F 8E 00 7F CE 04 00 6F 01 6F 03 86 FF A7 00 6F 02 86 37 A7 03 86 3C A7 01 97 09 4F 97 07 97 04 97 05 97 06 97 08 0E 20 FE 16 48 48 48 1B CE 00 13 DF 0F CE FD 76 BD
FD 21 C6 09 7E FB 0A 96 1B B7 04 00 96 13 97 1C 96 14 97 1D DE 18 96 1C 73 04 00 09 27 10 4A 26 FA 73 04 00 96 1D 09 27 05 4A 26 FA 20 E8 B6 04 00 2B 01 43 8B 00 B7 04
00 96 1C 9B 15 97 1C 96 1D 9B 16 97 1D 91 17 26 CB 96 1A 27 06 9B 13 97 13 26 B9 39 86 01 97 1A C6 03 20 0A 86 FE 97 1A 86 C0 C6 10 20 00 97 19 86 FF B7 04 00 D7 15 D6
15 96 0A 44 44 44 98 0A 44 76 00 09 76 00 0A 24 03 73 04 00 96 19 4A 26 FD 5A 26 E5 96 19 9B 1A 97 19 26 DB 39 86 20 97 15 97 18 86 01 CE 00 01 C6 FF 20 00 97 13 DF 16
D7 14 D6 15 96 0A 44 44 44 98 0A 44 76 00 09 76 00 0A 86 00 24 02 96 14 B7 04 00 DE 16 09 26 FD 5A 26 E1 D6 14 D0 13 27 09 DE 16 08 96 18 27 D0 20 CC 39 C6 01 D7 04 4F
97 19 20 14 4F 97 19 C6 03 20 0D 86 01 97 19 CE 03 E8 86 01 C6 FF 20 00 97 18 D7 13 DF 16 7F 00 15 DE 16 B6 04 00 16 54 54 54 D8 0A 54 76 00 09 76 00 0A D6 13 7D 00 19
27 02 D4 09 D7 14 D6 15 91 0A 22 12 09 27 26 B7 04 00 DB 15 99 14 25 16 91 0A 23 F0 20 10 09 27 14 B7 04 00 D0 15 92 14 25 04 91 0A 22 F0 96 0A B7 04 00 20 B9 D6 18 27
B5 96 13 D6 15 44 56 44 56 44 56 43 50 82 FF DB 15 99 13 D7 15 97 13 26 98 C1 07 26 94 39 86 FD 97 0F CE 00 64 DF 0B DB 0C 96 11 99 0B 97 11 DE 0B 25 04 20 00 20 03 08
27 11 DF 0B 84 0F 8B 9A 97 10 DE 0F A6 00 B7 04 00 20 DC 39 4F B7 04 00 97 11 4F 91 11 26 03 73 04 00 C6 12 5A 26 FD 4C 2A F1 73 04 00 7C 00 11 2A E8 39 CE 00 13 6F 00
08 8C 00 1B 26 F8 86 40 97 13 CE 00 13 86 80 97 11 5F A6 01 AB 00 A7 01 2A 02 DB 11 74 00 11 08 08 8C 00 1B 26 EC F7 04 00 7C 00 12 26 DC CE 00 13 5F A6 00 27 0B 81 37
26 04 C6 41 E7 02 6A 00 5C 08 08 8C 00 1B 26 EA 5D 26 BF 39 7A 00 08 39 7F 00 08 97 11 CE FD AA A6 00 27 2D 7A 00 11 27 06 4C BD FD 21 20 F1 08 DF 0F BD FD 21 DF 0D DE
0F A6 00 97 15 A6 01 EE 02 DF 13 8D 3E DE 0F 08 08 08 08 DF 0F 9C 0D 26 E8 7E FD 0E 86 03 97 08 39 7A 00 08 27 0C D6 15 58 58 58 58 1B 97 15 4F 20 FE 4A 81 0B 23 01 4F
CE FE 41 BD FD 21 A6 00 CE FF FF DF 13 8D 04 8D 2A 20 FC CE 00 16 81 00 27 15 81 03 27 09 C6 01 E7 00 08 80 02 20 EF C6 91 E7 00 6F 01 08 08 C6 7E E7 00 C6 FA E7 01 C6
DD E7 02 DE 13 4F F6 00 12 5C D7 12 D4 15 54 89 00 54 89 00 54 89 00 54 89 00 54 89 00 54 89 00 54 89 00 54 89 00 1B 48 48 48 48 B7 04 00 09 27 03 7E 00 16 39 36 A6 00 DF 0D DE
0F A7 00 08 DF 0F DE 0D 08 5A 26 EF 32 39 4F 97 04 97 05 39 7F 00 04 96 05 84 7F 81 1D 26 01 4F 4C 97 05 39 86 0E BD FB 81 96 05 48 48 43 BD FC 39 7C 00 17 BD FC 3B 20
F8 86 03 BD F8 2A D6 06 C1 1F 26 01 5F 5C D7 06 86 20 10 5F 81 14 23 05 CB 0E 4A 20 F7 CB 05 4A 26 FB D7 13 BD F8 3F 20 FB 96 07 26 09 7C 00 07 86 0D 8D 05 20 69 7E FC
2E 16 58 1B 1B 1B CE FE EC BD FD 21 A6 00 16 84 0F 97 14 54 54 54 54 D7 13 A6 01 16 54 54 54 54 D7 15 84 0F 97 11 DF 0B CE FE 4D 7A 00 11 2B 08 A6 00 4C BD FD 21 20 F3
DF 18 BD FC 75 DE 0B A6 02 97 1A BD FC 87 DE 0B A6 03 97 16 A6 04 97 17 A6 05 16 A6 06 CE FF 55 BD FD 21 17 DF 1B 7F 00 23 BD FD 21 DF 1D 39 96 13 97 22 DE 1B DF 0D DE
0D A6 00 9B 23 97 21 9C 1D 27 26 D6 14 08 DF 0D CE 00 24 96 21 4A 26 FD A6 00 B7 04 00 08 9C 1F 26 F1 5A 27 DA 08 09 08 09 08 09 08 09 01 01 20 DF 96 15 8D 62 7A 00 22
26 C1 96 07 26 46 96 16 27 42 7A 00 17 27 3D 9B 23 97 23 DE 1B 5F 96 23 7D 00 16 2B 06 AB 00 25 08 20 0B AB 00 27 02 25 05 5D 27 08 20 0F 5D 26 03 DF 1B 5C 08 9C 1D 26
DD 5D 26 01 39 DF 1D 96 15 27 06 8D 08 96 1A 8D 16 7E FB E7 39 CE 00 24 DF 0F DE 18 E6 00 08 BD FB 0A DE 0F DF 1F 39 4D 27 2B DE 18 DF 0D CE 00 24 97 12 DF 0F DE 0D D6
12 D7 11 E6 01 54 54 54 54 08 DF 0D DE 0F A6 00 10 7A 00 11 26 FA A7 00 08 9C 1F 26 DE 39 8E 00 7F B6 04 02 0E 43 84 1F D6 08 27 09 2A 03 BD FA 48 4A BD FA 89 5F 81 0E
27 02 D7 06 81 12 27 02 D7 07 F6 EF FD C1 7E 26 03 BD EF FD 4D 27 27 4A 81 0C 22 08 BD FB 81 BD FB E7 20 1A 81 1B 22 0E 80 0D 48 CE FD 58 8D 21 EE 00 AD 00 20 08 80 1C
BD F8 2A BD F8 3F 96 04 9A 05 27 FE 4F 97 07 96 04 27 03 7E F9 13 7E FB 34 DF 0D 9B 0E 97 0E 24 03 7C 00 0D DE 0D 39 0F 8E 00 7F CE FF FF 5F E9 00 09 8C F8 00 26 F8 E1
00 27 01 3E 86 01 BD F8 2A BD F8 3F F6 EF FA C1 7E 26 DC BD EF FA 20 D7 FB 49 F9 13 FB 24 F8 8C FB 71 FB 1E F8 CD F8 94 F9 1C F9 23 F9 A6 F9 D4 F9 F3 FA 44 FA 84 40 01
00 10 E1 00 80 FF FF 28 01 00 08 81 02 00 FF FF 28 81 00 FC 01 02 00 FC FF FF 01 00 18 41 04 80 00 FF 8C 5B B6 40 BF 49 A4 73 73 A4 49 BF 40 B6 5B 8C 0C 7F 1D 0F FB 7F
23 0F 15 FE 08 50 8B 88 3E 3F 02 3E 7C 04 03 FF 3E 3F 2C E2 7C 12 0D 74 7C 0D 0E 41 7C 23 0B 50 7C 1D 29 F2 7C 3F 02 3E F8 04 03 FF 7C 3F 2C E2 F8 12 0D 74 F8 0D 0E 41
F8 23 0B 50 F8 1D 2F F2 F8 23 05 A8 F8 12 06 BA F8 04 07 FF 7C 37 04 C1 7C 23 05 A8 7C 12 06 BA 3E 04 07 FF 3E 37 04 C1 3E 23 05 A8 1F 12 06 BA 1F 04 07 FF 1F 37 04 C1
1F 23 16 A0 FE 1D 17 F9 7F 37 13 06 7F 3F 08 FA FE 04 0F FF FE 0D 0E 41 FE 23 0B 50 FE 1D 5F E4 00 47 3F 37 30 29 23 1D 17 12 0D 08 04 08 7F D9 FF D9 7F 24 00 24 08 00
40 80 00 FF 00 80 40 10 7F B0 D9 F5 FF F5 D9 B0 7F 4E 24 09 00 09 24 4E 10 7F C5 EC E7 BF 8D 6D 6A 7F 94 92 71 40 17 12 39 10 FF FF FF FF 00 00 00 00 FF FF FF FF 00 00
00 00 48 8A 95 A0 AB B5 BF C8 D1 DA E1 E8 EE F3 F7 FB FD FE FF FE FD FB F7 F3 EE E8 E1 DA D1 C8 BF B5 AB A0 95 8A 7F 75 6A 5F 54 4A 40 37 2E 25 1E 17 11 0C 08 04 02 01
00 01 02 04 08 0C 11 17 1E 25 2E 37 40 4A 54 5F 6A 75 7F 10 59 7B 98 AC B3 AC 98 7B 59 37 19 06 00 06 19 37 81 24 00 00 00 16 31 12 05 1A FF 00 27 6D 11 05 11 01 0F 01
47 11 31 00 01 00 0D 1B F4 12 00 00 00 14 47 41 45 00 00 00 0F 5B 21 35 11 FF 00 0D 1B 15 00 00 FD 00 01 69 31 11 00 01 00 03 6A 01 15 01 01 01 01 47 F6 53 03 00 02 06
94 6A 10 02 00 02 06 9A 1F 12 00 FF 10 04 69 31 11 00 FF 00 0D 00 12 06 00 FF 01 09 28 A0 98 90 88 80 78 70 68 60 58 50 44 40 01 01 02 02 04 04 08 08 10 10 30 60 C0 E0
01 01 02 02 03 04 05 06 07 08 09 0A 0C 80 7C 78 74 70 74 78 7C 80 01 01 02 02 04 04 08 08 10 20 28 30 38 40 48 50 60 70 80 A0 B0 C0 08 40 08 40 08 40 08 40 08 40 08 40
08 40 08 40 08 40 08 40 01 02 04 08 09 0A 0B 0C 0E 0F 10 12 14 16 40 10 08 01 01 01 01 01 02 02 03 03 04 04 05 06 08 0A 0C 10 14 18 20 30 40 50 40 30 20 10 0C 0A 08 07
06 05 04 03 02 02 01 01 01 07 08 09 0A 0C 08 17 18 19 1A 1B 1C 00 00 00 FC B6 F8 01 FD 2F F8 01
```

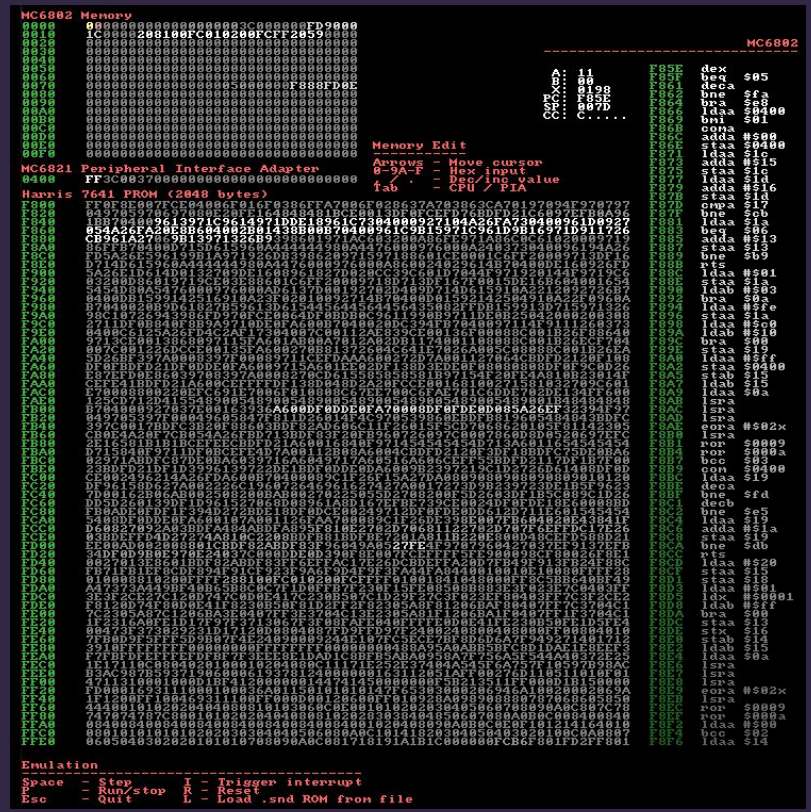
what is this madness?

Defender Sound ROM Map

Can we see this while the game is running?

defendemu

- Wrote Defender sound board emulator in 2015
- Emulated just enough of the MC6800 instruction set for a few sounds to play
- Active memory is highlighted



What was the Defender sound designer thinking?

Idea / Constraints

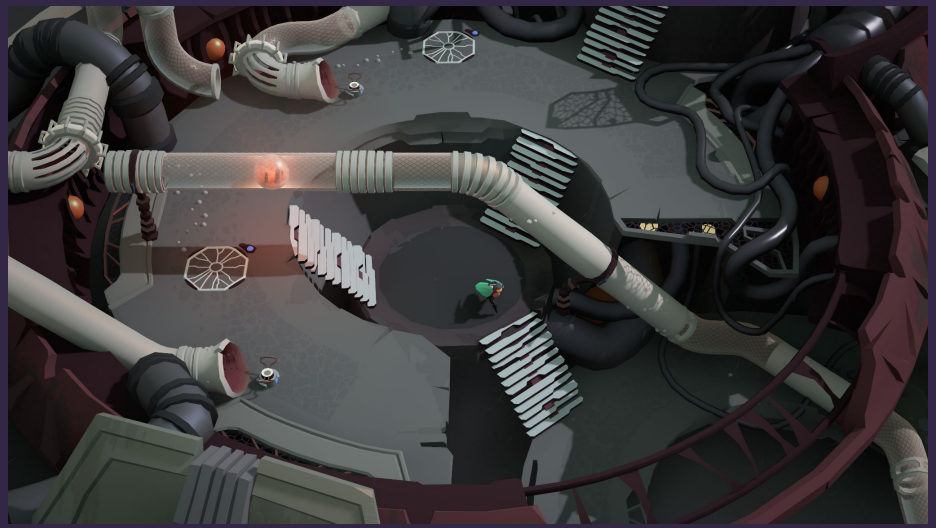
On this game the memory was only 512 bytes for all program and data. It was this extreme memory crunch that inspired the Gwave wave table synthesizer. By storing a waveform (sine, square, triangle,etc.) in 4-64 bytes, and then a frequency table of 10-20 bytes, a sound could be characterized by a few bytes. To get further mileage, echo, distortion, LFO,and white noise systems were also employed at a cost of only a few extra bytes.

[..] the most brilliant sounds were often created by typing in random numbers for the parameters. Often incredible sounds were generated by inputting mathematically undefined values, such as echoing a sound “0” times. The crudeness and lack of bounds checking of the program allowed for mathematical wraparound and error accumulation that sounded ethereal.

How did Defender inspire me personally?

COCOON

- Synthesized ambient music
- Was this even possible?
- Defender was the main inspiration



What were my thoughts at the time?

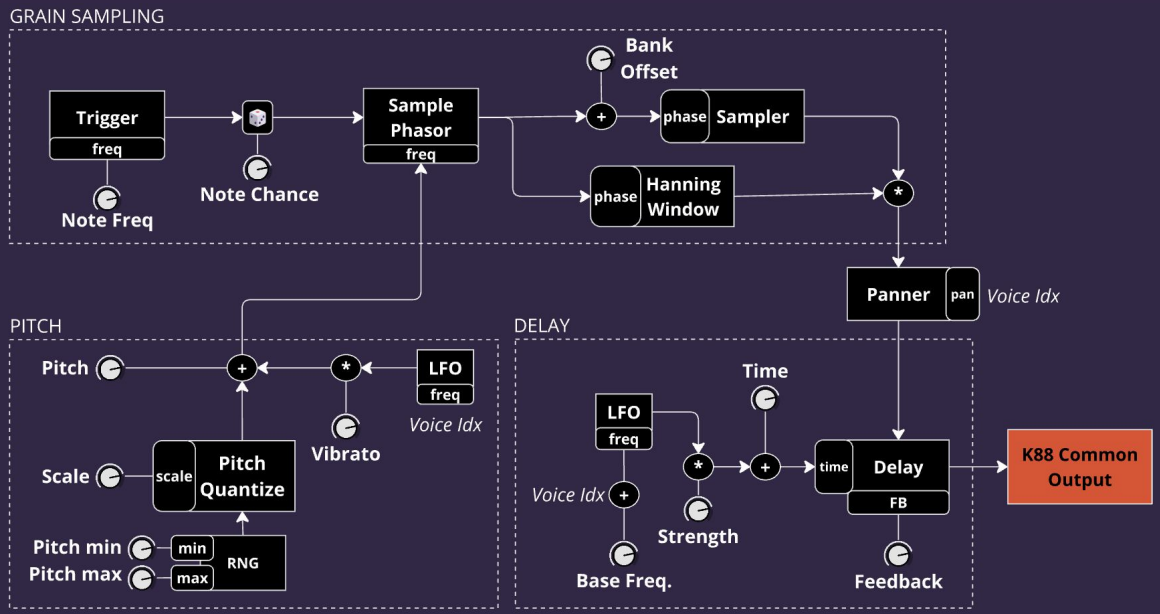
Reverse Idea/Constraints Dialogue

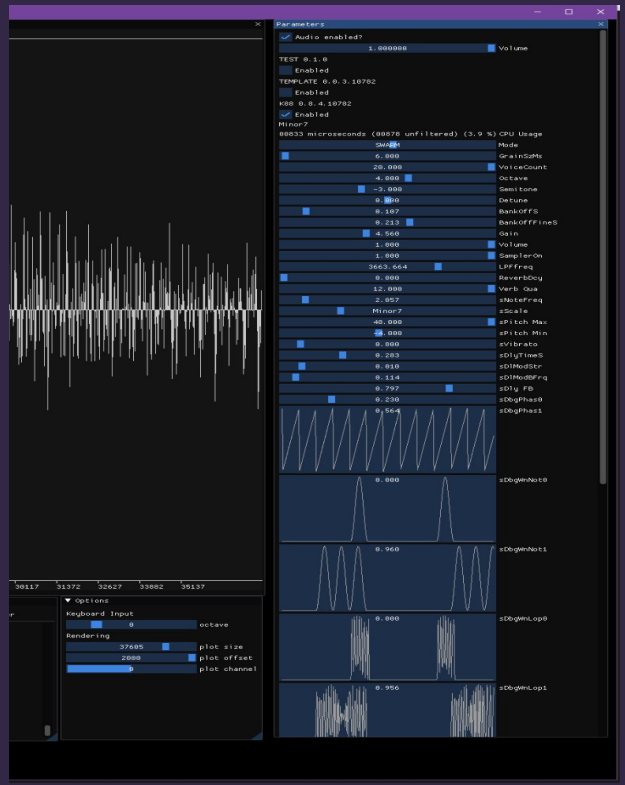
- My constraints were chosen willingly
- The ideas came out of the constraints
- Constraints led to artistic framework for music and sound design

COCOON Artistic Framework

Ambient music	Real-time synthesized ambient music for puzzle gameplay
Vignettes	Pre-rendered synthetic music vignettes for big moments
Sound design	Pre-rendered synthetic sounds for all sound design

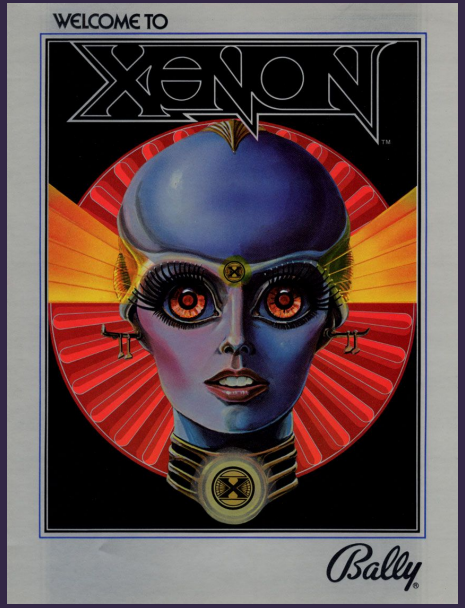
COCOON Synth Development





Sampling

- Sample ROMs unrealistic during the 1970s due to cost
- From around 1980, expensive hardware was able to support sampling



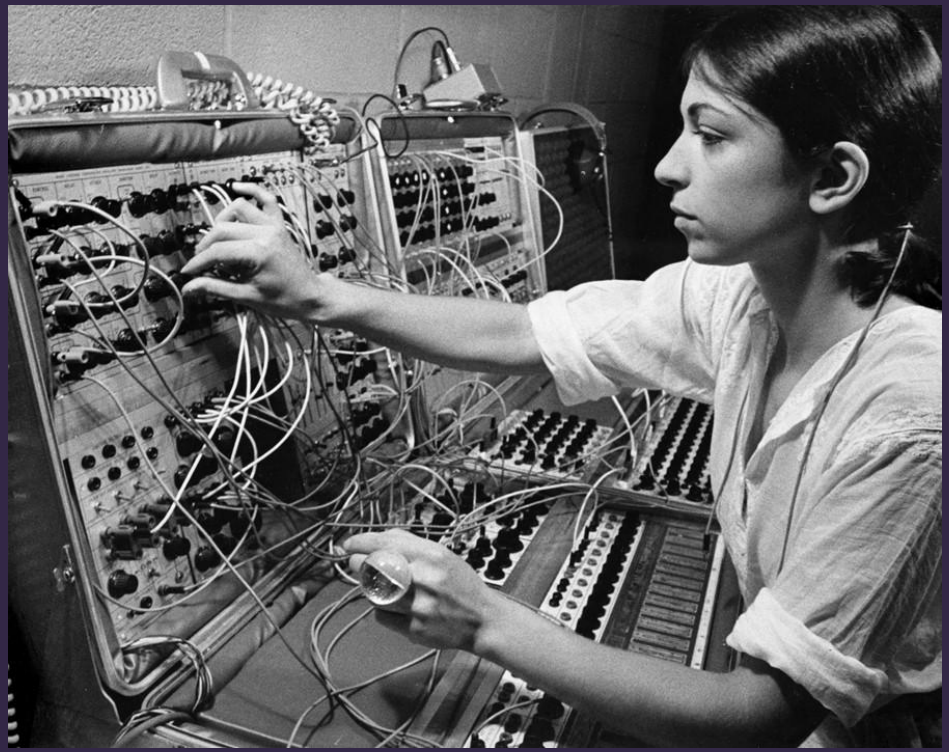

```
Title     : Xenon
Year      : 1980
Platform  : Pinball
Developer : Bally
Sound     : Samples, AY-3-8910,
            mechanical
```

What is Xenon?

Xenon

Very ambitious pinball machine project by Bally Manufacturing.

Sound design and music by
Suzanne Ciani



▸ gap26-ciani

Let's see the finished game!

Three Different Audio Types

Xenon had 3 different types of audio tech:

- MC6802 and 7 ROM chips (40 KB in total including other data) for voice samples and sampled adaptive music
- Programmable Sound Generator (AY-3-8910) for sound effects
- Physical mechanical sounds: ball, flippers, bumpers

Idea / Constraints

The idea of using short grunts and groans came to me when I watched people play the game. The way people expressed their frustrations or their involvement with the game, and I wanted the game to [..] talk back to the people playing.

The sound was designed to support the emotional levels and to be responsive to all the pinball gestures. [..] I saw that players were "performing" the game and thus the music!

The biggest challenge was how small the chip was, but limitations always inspire solutions. I was also the first female voice in a pinball game [..]

Inventing Voice Acting

What if you want voice acting but cheap out on sampling hardware?

Inventing Voice Acting

What if you want voice acting but cheap out on sampling hardware?


```
Title     : Shark Attack
Year      : 1980
Platform  : Arcade
Developer : Pacific Novelty
Sound     : 2 x AY-3-8910 + ?
```

Tapes

You stick a car stereo cassette tape player with auto reverse into your game

and enable/disable it from game logic

Idea / Constraints Dialogue

In the 1970-1980s game developers invented new types of audio for the medium:

- sound effects
- adaptive and interactive music
- ambience
- voice acting

Idea / Constraints Dialogue

The era imposed harsh time and hardware constraints on developers

But still, we saw developers:

- invent quick, effective solutions (Pong, Space Invaders)
- create new sound hardware from scratch (Carnival, Space Invaders)
- use/abuse limited hardware to create realistic sounds and great music (AY-3-8910, C64 SID)
- experiment with quirks, errors and limitations to create new sounds (Defender)
- create a sonic dialogue between human and machine (Xenon)
- stick a cassette player in an arcade cabinet (Shark Attack)

1980 vs. 2026

1980

Huge variety in sound reproduction technology

2026

Homogeneity in sound reproduction technology:

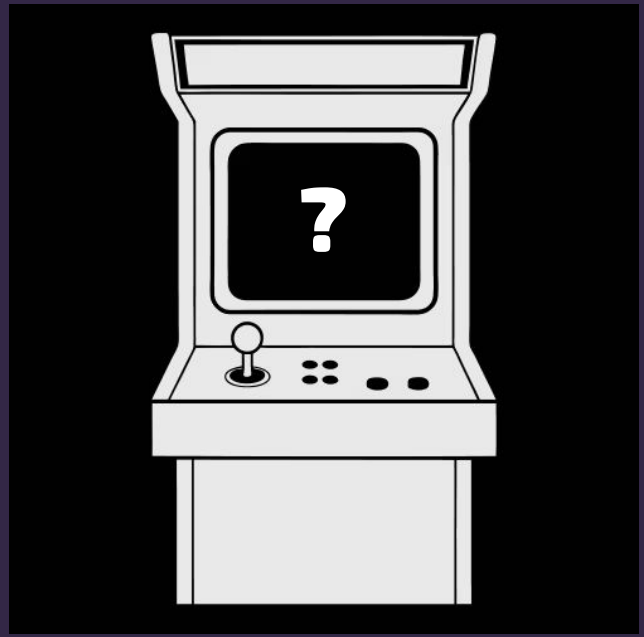
Digital software synthesis
(e.g. Wwise, FMOD, etc.)

However, huge variety in sound production techniques

Thought Experiment

So, it's 1980 and you're designing an arcade machine. What hardware are you using for sound?

- Mechanical sound
- Digital hardware synthesis
- Analog synthesis
- Programmable sound generators
- Software synthesis
- ROM samples
- Tapes

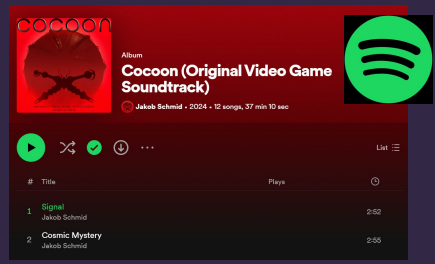


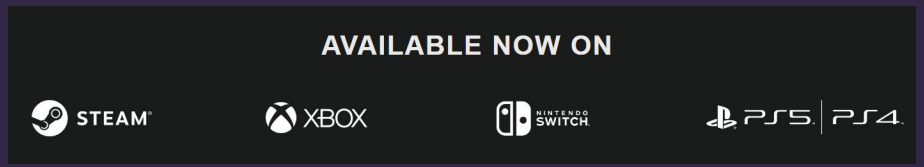
What will you choose?

Questions?

Thank you!

Web	cocoongame.com
E-mail	jakob@schmid.dk
Bluesky	@schmid.dk
x.com	@jakobschmid
Slides will be here	schmid.dk/talks



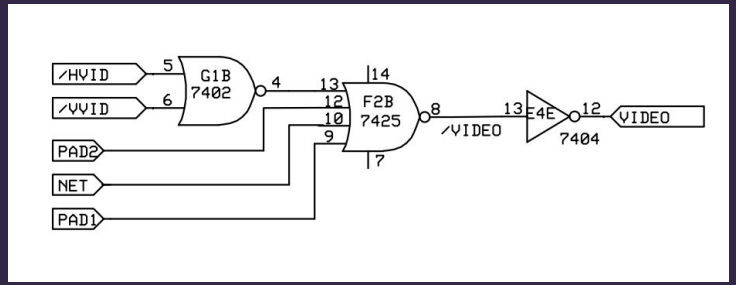


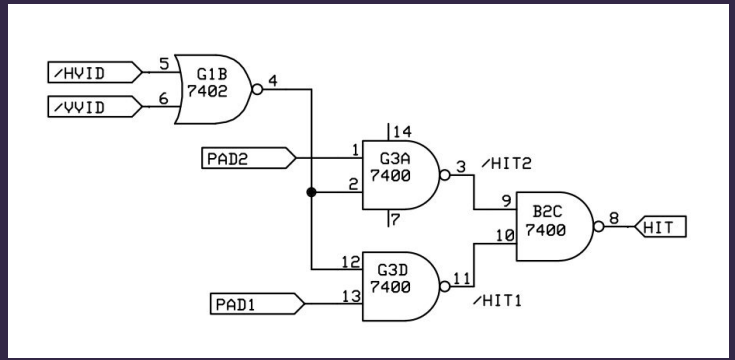
schmid-gap26-2026-01-21-1519.pdf

HIT SOUND: Ball-Bat Collision

What will be rendered where the electron gun is located?
/HVID: inside ball pixel
/VVID: inside ball pixel
PAD2: Inside player 2 paddle pixel
NET: Inside net pixel
PAD1: Inside player 1 paddle pixel

Are we inside the ball? (/HVID and /VVID)
Are we inside a paddle? (PAD2 or PAD1)
If both are in the same pixel, we have a HIT

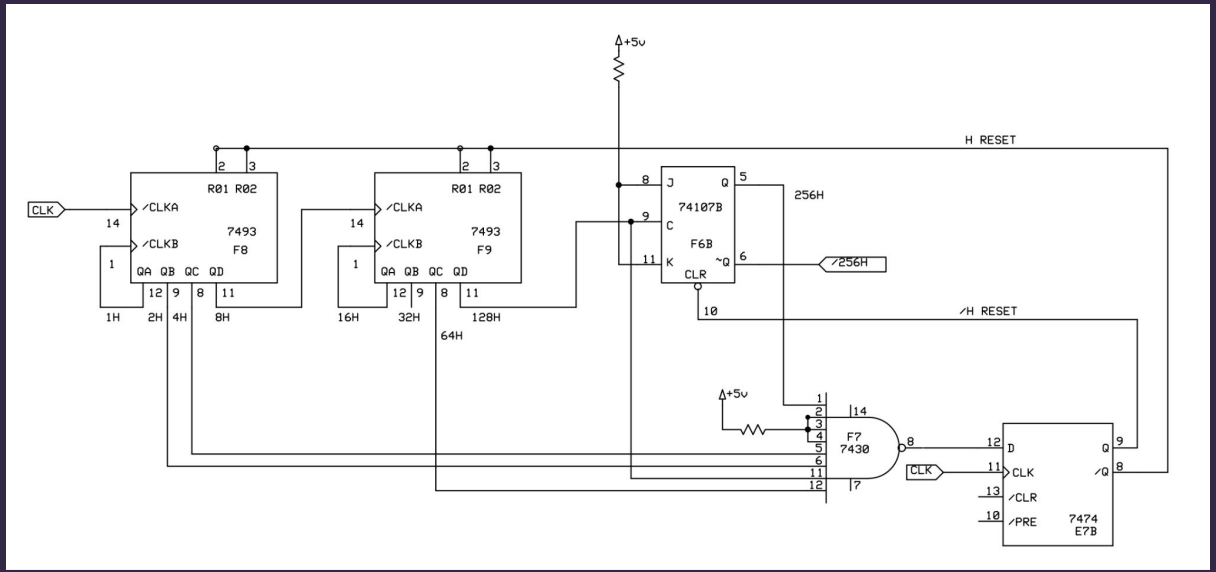




HSYNC

CLK is used as input for two 7493 4-bit counters and a 74107 flip flop, acts as 9-bit counter
Counts up to 454, then resets and sends HRESET, which later is used to generate HSYNC
HRESET occurs every 455 CLKs ~ 15.73 KHz

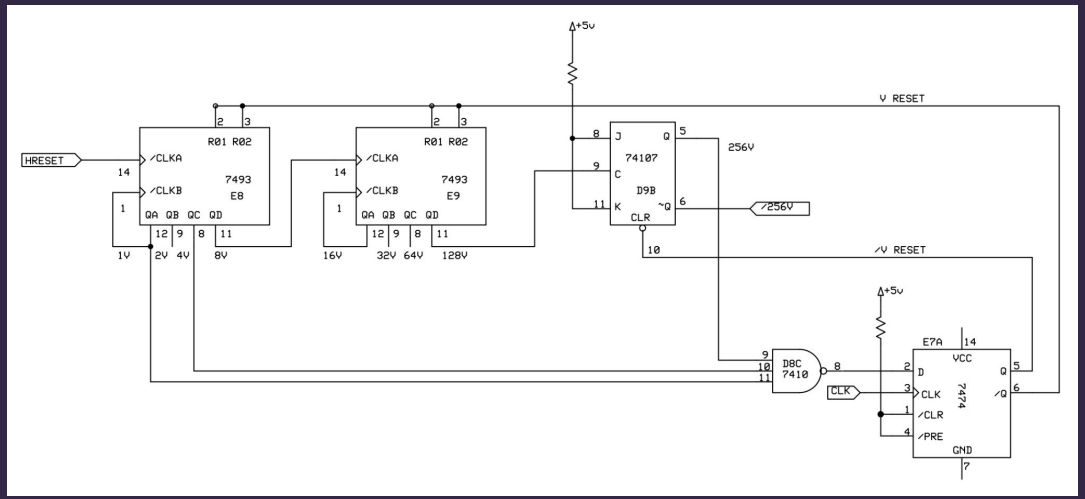
What about VSYNC?



VSYNC

HRESET is used as input for another set of 4-bit counters and a flip flop, acts as 9-bit counter
Counts up to 261, then resets and sends VRESET, which later is used to generate VSYNC
It also outputs 32V, which generates the SCORE sound

So what does 32V look like?



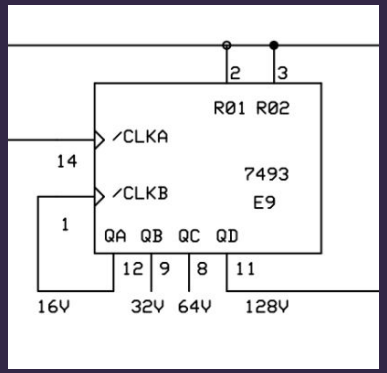
32V

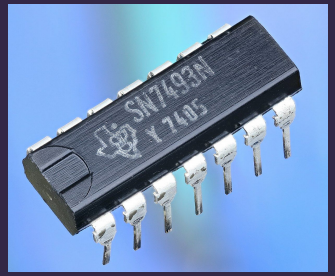
HRESET is a sequence of pulses
The 7493 chip counts, but we use the 2nd binary digit QB, which is a square wave
The frequency of 32V is F(HRESET)/64 = 245.8 Hz

Is it a perfect square wave?

'93A, 'L93, 'LS93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

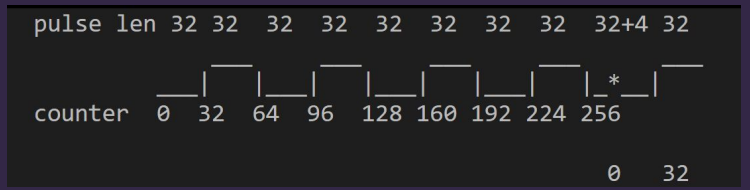




32V

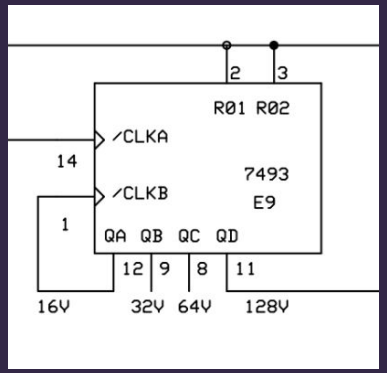
HRESET is a sequence of pulses
The 7493 chip counts, but we use the 2nd binary digit QB, which is a square wave
The frequency of 32V is F(HRESET)/64 = 245.8 Hz
It resets after 262 CLKs, making it glitchy

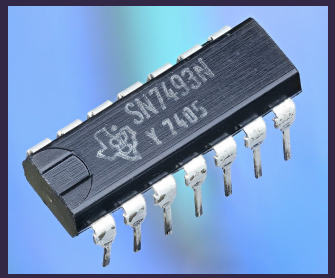
So, to summarize ...



'93A, 'L93, 'LS93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

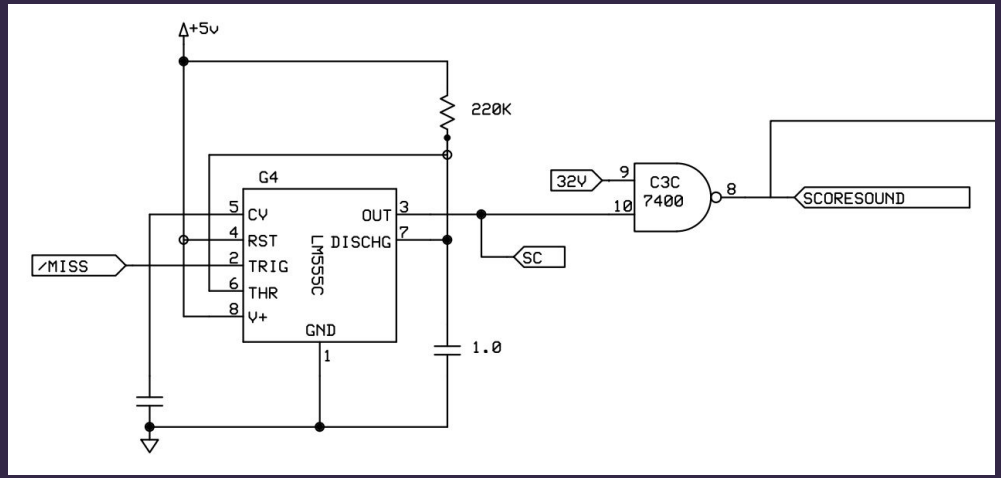




SCORE Sound

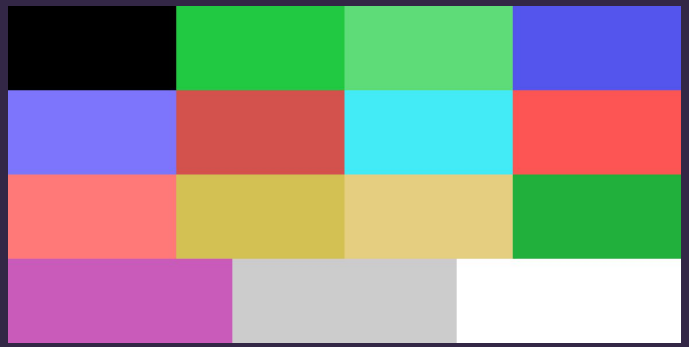
What is the 32V signal?

A glitchy 245.8 Hz square wave.



TMS9918

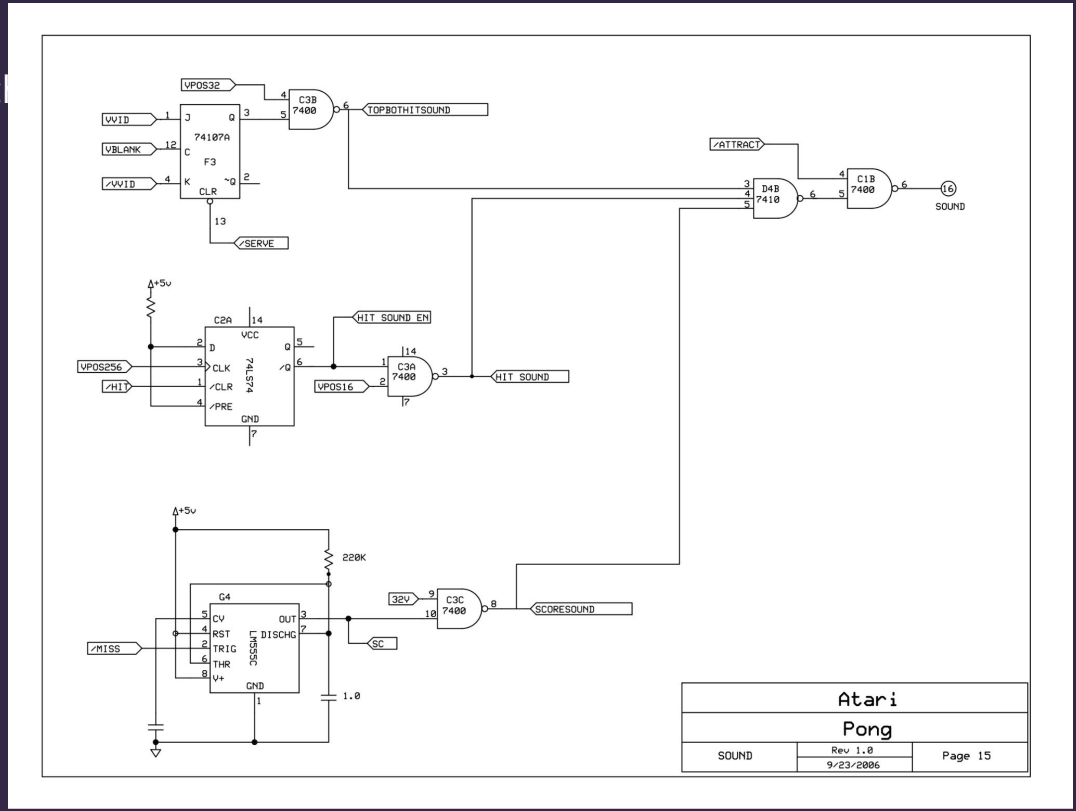
256x192 resolution with 15-color fixed palette
32 single-color sprites




```
Title     : Sega-Galaga
Platform  : SG-1000
Year      : 1984
Developer : Namco (port by Sega)
```

Sound Circuits

- We see 3 sound circuits, one for eac
- sound
- Digitally mixed together
- Muted by ATTRACT mode
- Sent directly to loudspeaker



Let's look at SCORE sound...

Defender Boot Sound, 98 bytes

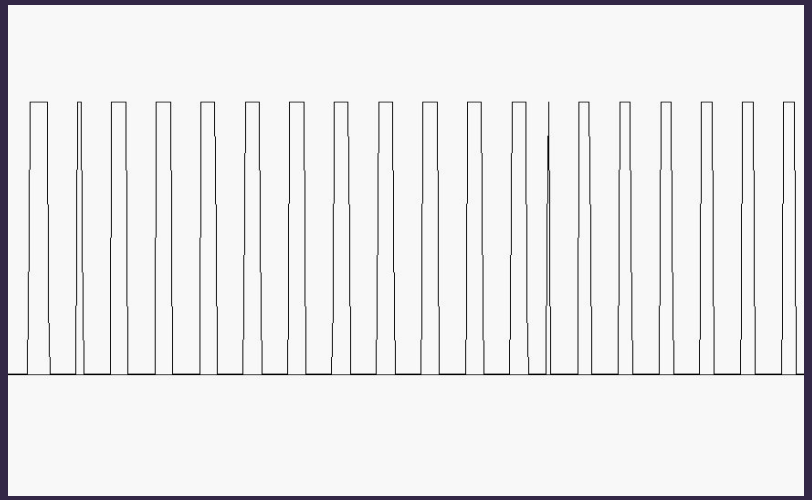
```
FF 0F 8E 00 7F CE 04 00 6F 01 6F 03 86 FF A7 00 6F 02 86 37 A7 03 86 3C A7 01 97 09 4F 97 07 97 04 97 05 97 06 97 08 0E 20 FE 16 48 48 48 1B CE 00 13 DF 0F CE FD 76 BD
FD 21 C6 09 7E FB 0A 96 1B B7 04 00 96 13 97 1C 96 14 97 1D DE 18 96 1C 73 04 00 09 27 10 4A 26 FA 73 04 00 96 1D 09 27 05 4A 26 FA 20 E8 B6 04 00 2B 01 43 8B 00 B7 04
00 96 1C 9B 15 97 1C 96 1D 9B 16 97 1D 91 17 26 CB 96 1A 27 06 9B 13 97 13 26 B9 39 86 01 97 1A C6 03 20 0A 86 FE 97 1A 86 C0 C6 10 20 00 97 19 86 FF B7 04 00 D7 15 D6
15 96 0A 44 44 44 98 0A 44 76 00 09 76 00 0A 24 03 73 04 00 96 19 4A 26 FD 5A 26 E5 96 19 9B 1A 97 19 26 DB 39 86 20 97 15 97 18 86 01 CE 00 01 C6 FF 20 00 97 13 DF 16
D7 14 D6 15 96 0A 44 44 44 98 0A 44 76 00 09 76 00 0A 86 00 24 02 96 14 B7 04 00 DE 16 09 26 FD 5A 26 E1 D6 14 D0 13 27 09 DE 16 08 96 18 27 D0 20 CC 39 C6 01 D7 04 4F
97 19 20 14 4F 97 19 C6 03 20 0D 86 01 97 19 CE 03 E8 86 01 C6 FF 20 00 97 18 D7 13 DF 16 7F 00 15 DE 16 B6 04 00 16 54 54 54 D8 0A 54 76 00 09 76 00 0A D6 13 7D 00 19
27 02 D4 09 D7 14 D6 15 91 0A 22 12 09 27 26 B7 04 00 DB 15 99 14 25 16 91 0A 23 F0 20 10 09 27 14 B7 04 00 D0 15 92 14 25 04 91 0A 22 F0 96 0A B7 04 00 20 B9 D6 18 27
B5 96 13 D6 15 44 56 44 56 44 56 43 50 82 FF DB 15 99 13 D7 15 97 13 26 98 C1 07 26 94 39 86 FD 97 0F CE 00 64 DF 0B DB 0C 96 11 99 0B 97 11 DE 0B 25 04 20 00 20 03 08
27 11 DF 0B 84 0F 8B 9A 97 10 DE 0F A6 00 B7 04 00 20 DC 39 4F B7 04 00 97 11 4F 91 11 26 03 73 04 00 C6 12 5A 26 FD 4C 2A F1 73 04 00 7C 00 11 2A E8 39 CE 00 13 6F 00
08 8C 00 1B 26 F8 86 40 97 13 CE 00 13 86 80 97 11 5F A6 01 AB 00 A7 01 2A 02 DB 11 74 00 11 08 08 8C 00 1B 26 EC F7 04 00 7C 00 12 26 DC CE 00 13 5F A6 00 27 0B 81 37
26 04 C6 41 E7 02 6A 00 5C 08 08 8C 00 1B 26 EA 5D 26 BF 39 7A 00 08 39 7F 00 08 97 11 CE FD AA A6 00 27 2D 7A 00 11 27 06 4C BD FD 21 20 F1 08 DF 0F BD FD 21 DF 0D DE
0F A6 00 97 15 A6 01 EE 02 DF 13 8D 3E DE 0F 08 08 08 08 DF 0F 9C 0D 26 E8 7E FD 0E 86 03 97 08 39 7A 00 08 27 0C D6 15 58 58 58 58 1B 97 15 4F 20 FE 4A 81 0B 23 01 4F
CE FE 41 BD FD 21 A6 00 CE FF FF DF 13 8D 04 8D 2A 20 FC CE 00 16 81 00 27 15 81 03 27 09 C6 01 E7 00 08 80 02 20 EF C6 91 E7 00 6F 01 08 08 C6 7E E7 00 C6 FA E7 01 C6
DD E7 02 DE 13 4F F6 00 12 5C D7 12 D4 15 54 89 00 54 89 00 54 89 00 54 89 00 54 89 00 54 89 00 54 89 00 1B 48 48 48 B7 04 00 09 27 03 7E 00 16 39 36 A6 00 DF 0D DE
0F A7 00 08 DF 0F DE 0D 08 5A 26 EF 32 39 4F 97 04 97 05 39 7F 00 04 96 05 84 7F 81 1D 26 01 4F 4C 97 05 39 86 0E BD FB 81 96 05 48 48 43 BD FC 39 7C 00 17 BD FC 3B 20
F8 86 03 BD F8 2A D6 06 C1 1F 26 01 5F 5C D7 06 86 20 10 5F 81 14 23 05 CB 0E 4A 20 F7 CB 05 4A 26 FB D7 13 BD F8 3F 20 FB 96 07 26 09 7C 00 07 86 0D 8D 05 20 69 7E FC
2E 16 58 1B 1B 1B CE FE EC BD FD 21 A6 00 16 84 0F 97 14 54 54 54 54 D7 13 A6 01 16 54 54 54 54 D7 15 84 0F 97 11 DF 0B CE FE 4D 7A 00 11 2B 08 A6 00 4C BD FD 21 20 F3
DF 18 BD FC 75 DE 0B A6 02 97 1A BD FC 87 DE 0B A6 03 97 16 A6 04 97 17 A6 05 16 A6 06 CE FF 55 BD FD 21 17 DF 1B 7F 00 23 BD FD 21 DF 1D 39 96 13 97 22 DE 1B DF 0D DE
0D A6 00 9B 23 97 21 9C 1D 27 26 D6 14 08 DF 0D CE 00 24 96 21 4A 26 FD A6 00 B7 04 00 08 9C 1F 26 F1 5A 27 DA 08 09 08 09 08 09 08 09 01 01 20 DF 96 15 8D 62 7A 00 22
26 C1 96 07 26 46 96 16 27 42 7A 00 17 27 3D 9B 23 97 23 DE 1B 5F 96 23 7D 00 16 2B 06 AB 00 25 08 20 0B AB 00 27 02 25 05 5D 27 08 20 0F 5D 26 03 DF 1B 5C 08 9C 1D 26
DD 5D 26 01 39 DF 1D 96 15 27 06 8D 08 96 1A 8D 16 7E FB E7 39 CE 00 24 DF 0F DE 18 E6 00 08 BD FB 0A DE 0F DF 1F 39 4D 27 2B DE 18 DF 0D CE 00 24 97 12 DF 0F DE 0D D6
12 D7 11 E6 01 54 54 54 54 54 08 DF 0D DE 0F A6 00 10 7A 00 11 26 FA A7 00 08 9C 1F 26 DE 39 8E 00 7F B6 04 02 0E 43 84 1F D6 08 27 09 2A 03 BD FA 48 4A BD FA 89 5F 81 0E
27 02 D7 06 81 12 27 02 D7 07 F6 EF FD C1 7E 26 03 BD EF FD 4D 27 27 4A 81 0C 22 08 BD FB 81 BD FB E7 20 1A 81 1B 22 0E 80 0D 48 CE FD 58 8D 21 EE 00 AD 00 20 08 80 1C
BD F8 2A BD F8 3F 96 04 9A 05 27 FE 4F 97 07 96 04 27 03 7E F9 13 7E FB 34 DF 0D 9B 0E 97 0E 24 03 7C 00 0D DE 0D 39 0F 8E 00 7F CE FF FF 5F E9 00 09 8C F8 00 26 F8 E1
00 27 01 3E 86 01 BD F8 2A BD F8 3F F6 EF FA C1 7E 26 DC BD EF FA 20 D7 FB 49 F9 13 FB 24 F8 8C FB 71 FB 1E F8 CD F8 94 F9 1C F9 23 F9 A6 F9 D4 F9 F3 FA 44 FA 84 40 01
00 10 E1 00 80 FF FF 28 01 00 08 81 02 00 FF FF 28 81 00 FC 01 02 00 FC FF FF 01 00 18 41 04 80 00 FF 8C 5B B6 40 BF 49 A4 73 73 A4 49 BF 40 B6 5B 8C 0C 7F 1D 0F FB 7F
23 0F 15 FE 08 50 8B 88 3E 3F 02 3E 7C 04 03 FF 3E 3F 2C E2 7C 12 0D 74 7C 0D 0E 41 7C 23 0B 50 7C 1D 29 F2 7C 3F 02 3E F8 04 03 FF 7C 3F 2C E2 F8 12 0D 74 F8 0D 0E 41
F8 23 0B 50 F8 1D 2F F2 F8 23 05 A8 F8 12 06 BA F8 04 07 FF 7C 37 04 C1 7C 23 05 A8 7C 12 06 BA 3E 04 07 FF 3E 37 04 C1 3E 23 05 A8 1F 12 06 BA 1F 04 07 FF 1F 37 04 C1
1F 23 16 A0 FE 1D 17 F9 7F 37 13 06 7F 3F 08 FA FE 04 0F FF FE 0D 0E 41 FE 23 0B 50 FE 1D 5F E4 00 47 3F 37 30 29 23 1D 17 12 0D 08 04 08 7F D9 FF D9 7F 24 00 24 08 00
40 80 00 FF 00 80 40 10 7F B0 D9 F5 FF F5 D9 B0 7F 4E 24 09 00 09 24 4E 10 7F C5 EC E7 BF 8D 6D 6A 7F 94 92 71 40 17 12 39 10 FF FF FF FF 00 00 00 00 FF FF FF FF 00 00
00 00 48 8A 95 A0 AB B5 BF C8 D1 DA E1 E8 EE F3 F7 FB FD FE FF FE FD FB F7 F3 EE E8 E1 DA D1 C8 BF B5 AB A0 95 8A 7F 75 6A 5F 54 4A 40 37 2E 25 1E 17 11 0C 08 04 02 01
00 01 02 04 08 0C 11 17 1E 25 2E 37 40 4A 54 5F 6A 75 7F 10 59 7B 98 AC B3 AC 98 7B 59 37 19 06 00 06 19 37 81 24 00 00 00 16 31 12 05 1A FF 00 27 6D 11 05 11 01 0F 01
47 11 31 00 01 00 0D 1B F4 12 00 00 00 14 47 41 45 00 00 00 0F 5B 21 35 11 FF 00 0D 1B 15 00 00 FD 00 01 69 31 11 00 01 00 03 6A 01 15 01 01 01 01 47 F6 53 03 00 02 06
94 6A 10 02 00 02 06 9A 1F 12 00 FF 10 04 69 31 11 00 FF 00 0D 00 12 06 00 FF 01 09 28 A0 98 90 88 80 78 70 68 60 58 50 44 40 01 01 02 02 04 04 08 08 10 10 30 60 C0 E0
01 01 02 02 03 04 05 06 07 08 09 0A 0C 80 7C 78 74 70 74 78 7C 80 01 01 02 02 04 04 08 08 10 20 28 30 38 40 48 50 60 70 80 A0 B0 C0 08 40 08 40 08 40 08 40 08 40 08 40
08 40 08 40 08 40 08 40 01 02 04 08 09 0A 0B 0C 0E 0F 10 12 14 16 40 10 08 01 01 01 01 01 02 02 03 03 04 04 05 06 08 0A 0C 10 14 18 20 30 40 50 40 30 20 10 0C 0A 08 07
06 05 04 03 02 02 01 01 01 07 08 09 0A 0C 08 17 18 19 1A 1B 1C 00 00 00 FC B6 F8 01 FD 2F F8 01
```

Boot Sound Loop Disassembled

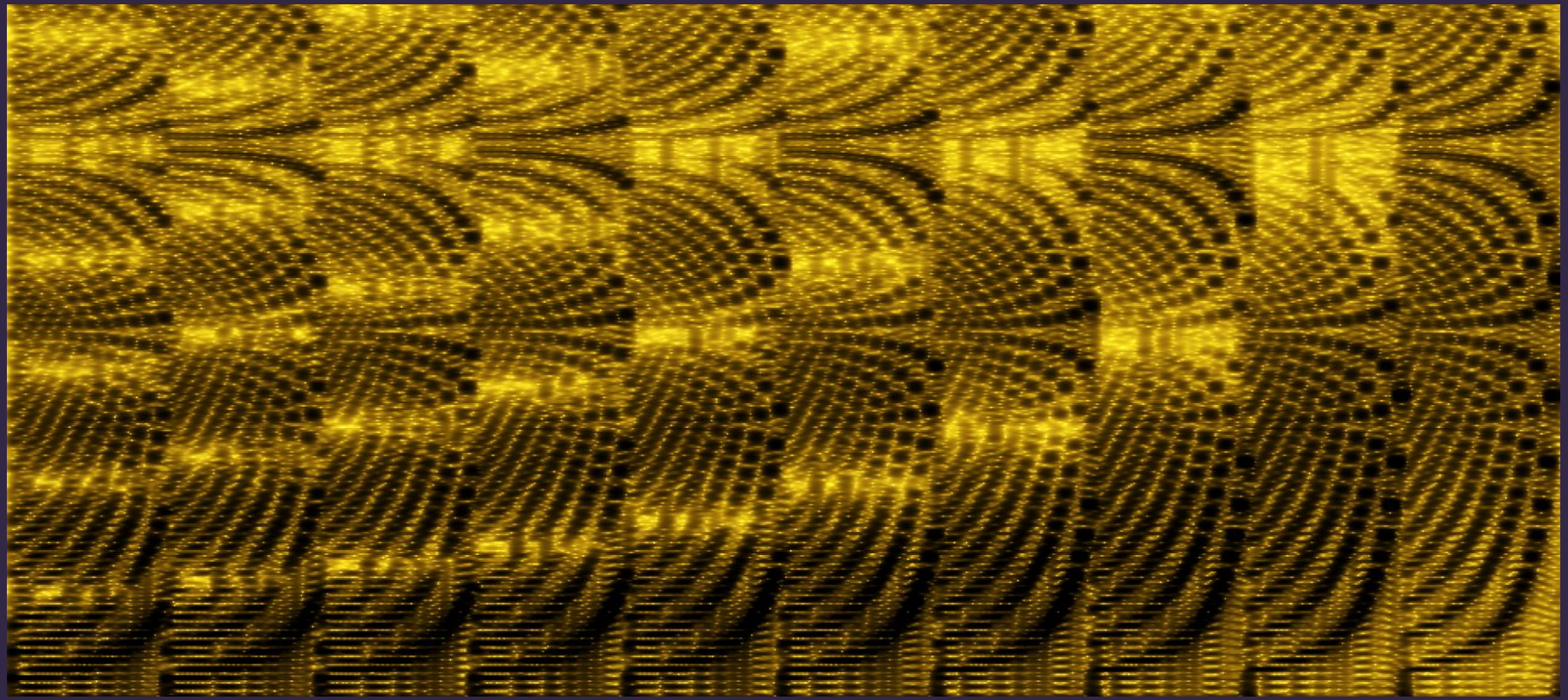
```
    lda  $1B      ; A = volume (always 255)
    sta  $0400    ; output A to DAC
loop:
    com  $0400    ; invert DAC output
                  ; - alternates between 0
                  ;   and 255

    ...           ; variable delay
                  ; i.e. pulse-width modulation

    bra  loop     ; goto loop
```



What We Hear



Sound in Marketing