

# Sega Game Gear on a Chip

Max Thrun — Samir Silbak

University of Cincinnati

Fall 2012

# First slide

some content here

# Sega Game Gear

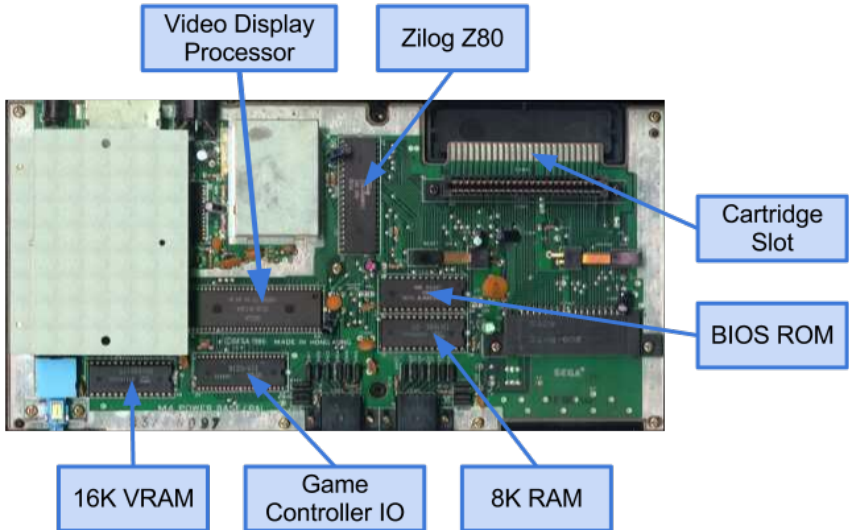


# Sega Game Gear

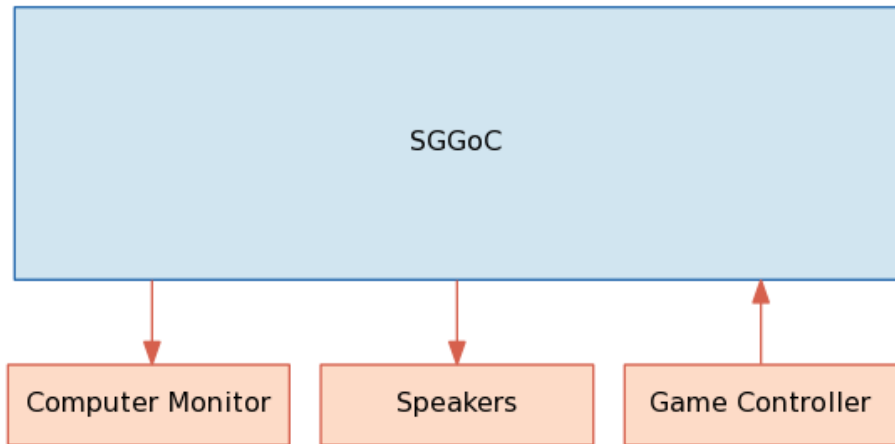
- Released April 1990
- Mobile version of the Sega Master System (functionally identical)
- Standard system architecture for the time (Z80 CPU, tri-state buses, etc...)



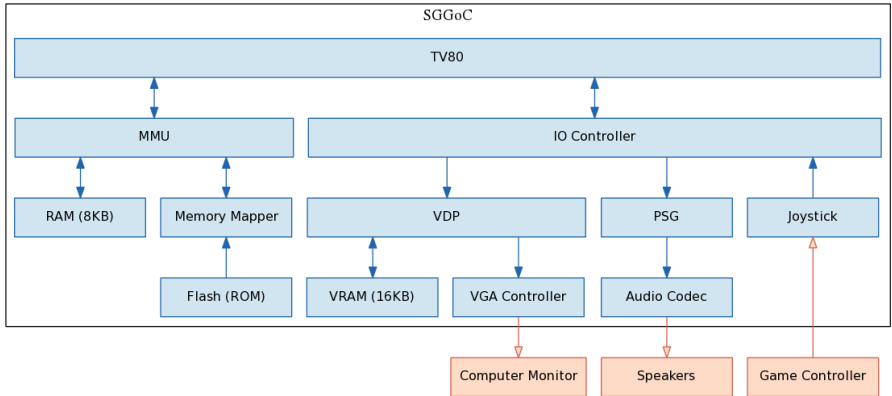
# Sega Master System PCB



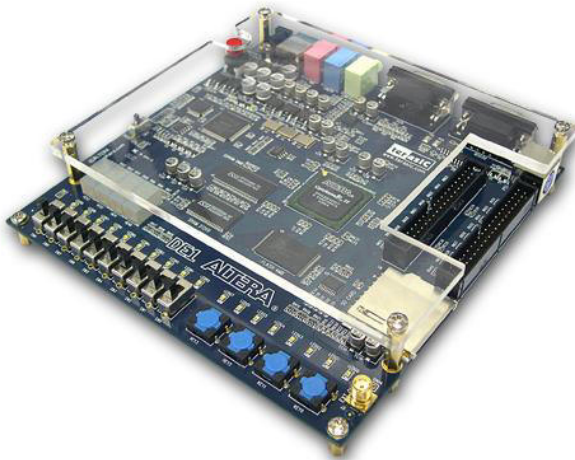
# Black Box Diagram



# Transparent Box Diagram



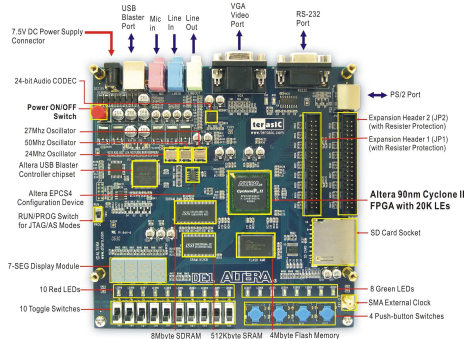
# FPGA Development Board





# Altera DE1

- Cyclone II EP2C20F484C7
- VGA, Audio, SD Card, 4 MB Flash
- Command line development environment
- Extremely good documentation



GAME GEAR™

**ECCO**  
THE TIDES OF TIME

Officially Licensed  
**SEGA**  
GAMES

2541

**SEGA**™

ESRB  
RATING  
GA  
General Audience  
Suitable for all audiences  
Content Descriptors



# Game Cartridge

Each game cartridge made up of at least two components:

- Game data ROM
- Memory Mapper



Practically every game ROM has been dumped as is available online

# Game Cartridge

Each game cartridge made up of at least two components:

- Game data ROM
- Memory Mapper



Practically every game ROM has been dumped as is available online

# Game Cartridge

Each game cartridge made up of at least two components:

- Game data ROM
- Memory Mapper

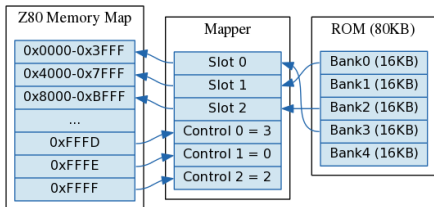


Practically every game ROM has been dumped as is available online

# Game Cartridge

Each game cartridge made up of at least two components:

- Game data ROM
- Memory Mapper

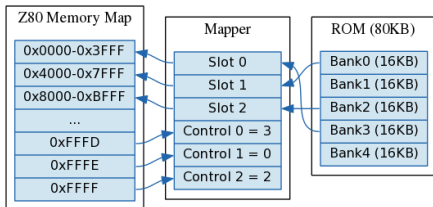


Practically every game ROM has been dumped as is available online

# Game Cartridge

Each game cartridge made up of at least two components:

- Game data ROM
- Memory Mapper



Practically every game ROM has been dumped as is available online

# Storing Game ROMs

A few options to store game ROMs:

1. Hookup the actual cartridge
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips



# Storing Game ROMs

A few options to store game ROMs:

1. Hookup the actual cartridge
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. Hookup the actual cartridge
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. Hookup the actual cartridge
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. Hookup the actual cartridge
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. Hookup the actual cartridge
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. **Hookup the actual cartridge**
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. **Hookup the actual cartridge**
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. **Hookup the actual cartridge**
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips



# Storing Game ROMs

A few options to store game ROMs:

1. **Hookup the actual cartridge**
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. **Store them on a SD card**
  - Extremely portable / convenient
  - Even more complicated
3. **Store them on the 4MB flash chip**
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. **Hookup the actual cartridge**
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. **Store them on a SD card**
  - Extremely portable / convenient
  - Even more complicated
3. **Store them on the 4MB flash chip**
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. **Hookup the actual cartridge**
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. **Store them on a SD card**
  - Extremely portable / convenient
  - Even more complicated
3. **Store them on the 4MB flash chip**
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. **Hookup the actual cartridge**
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. **Store them on a SD card**
  - Extremely portable / convenient
  - Even more complicated
3. **Store them on the 4MB flash chip**
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# Storing Game ROMs

A few options to store game ROMs:

1. Hookup the actual cartridge
  - Straight forward
  - Don't have to re-implement the memory mappers
  - Defeats most the point of the project
2. Store them on a SD card
  - Extremely portable / convenient
  - Even more complicated
3. Store them on the 4MB flash chip
  - Fairly straightforward
  - Extremely non-portable
  - Flash chip looks just like original ROM chips

# ROM Flasher Tool

Need a tool to load a ROM file into the flash chip from the PC

1. Load RS232-to-ROM bridge into the FPGA
2. PC waits for FPGA to request a byte
3. PC send the next byte of ROM file
4. FPGA writes byte to flash
5. Go back to 2

Can also do the reverse to read back and verify the flash contents against ROM file

# ROM Flasher Tool

Need a tool to load a ROM file into the flash chip from the PC

1. Load RS232-to-ROM bridge into the FPGA
2. PC waits for FPGA to request a byte
3. PC send the next byte of ROM file
4. FPGA writes byte to flash
5. Go back to 2

Can also do the reverse to read back and verify the flash contents against ROM file

# ROM Flasher Tool

Need a tool to load a ROM file into the flash chip from the PC

1. Load RS232-to-ROM bridge into the FPGA
2. PC waits for FPGA to request a byte
3. PC send the next byte of ROM file
4. FPGA writes byte to flash
5. Go back to 2

Can also do the reverse to read back and verify the flash contents against ROM file



# ROM Flasher Tool

Need a tool to load a ROM file into the flash chip from the PC

1. Load RS232-to-ROM bridge into the FPGA
2. PC waits for FPGA to request a byte
3. PC send the next byte of ROM file
4. FPGA writes byte to flash
5. Go back to 2

Can also do the reverse to read back and verify the flash contents against ROM file

# ROM Flasher Tool

Need a tool to load a ROM file into the flash chip from the PC

1. Load RS232-to-ROM bridge into the FPGA
2. PC waits for FPGA to request a byte
3. PC send the next byte of ROM file
4. FPGA writes byte to flash
5. Go back to 2

Can also do the reverse to read back and verify the flash contents against ROM file

# ROM Flasher Tool

Need a tool to load a ROM file into the flash chip from the PC

1. Load RS232-to-ROM bridge into the FPGA
2. PC waits for FPGA to request a byte
3. PC send the next byte of ROM file
4. FPGA writes byte to flash
5. Go back to 2

Can also do the reverse to read back and verify the flash contents against ROM file

# ROM Flasher Tool

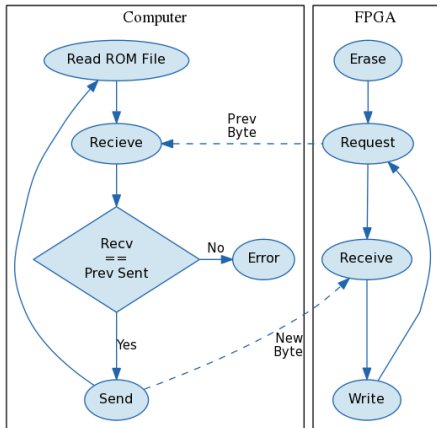
Need a tool to load a ROM file into the flash chip from the PC

1. Load RS232-to-ROM bridge into the FPGA
2. PC waits for FPGA to request a byte
3. PC send the next byte of ROM file
4. FPGA writes byte to flash
5. Go back to 2

Can also do the reverse to read back and verify the flash contents against ROM file

# ROM Flasher Tool

## Writing



## Reading

