



Elias De Smijter  
Erik Wilhelm  
Widlund Mellergård

## Mini-project: Audio volume control



5 june 2023



## OUTLINE

### Goal of the project

System description

System functioning

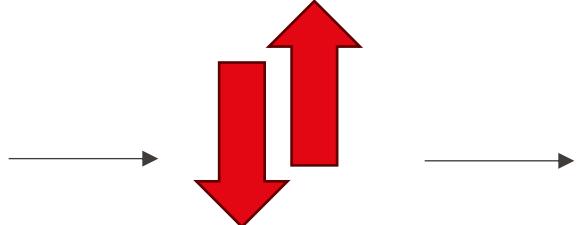
Profiling

Conclusion

Demo

# User-interface goals

- Record sound
- Change volume up/down
- Playback sound



# Hardware-related goals

- Dual-core
- Three methods
  - Hardware acceleration using DMA
  - Custom instruction
  - C instruction
- Last two not used in practice, used for profiling



## OUTLINE

Goal of the project

**System description**

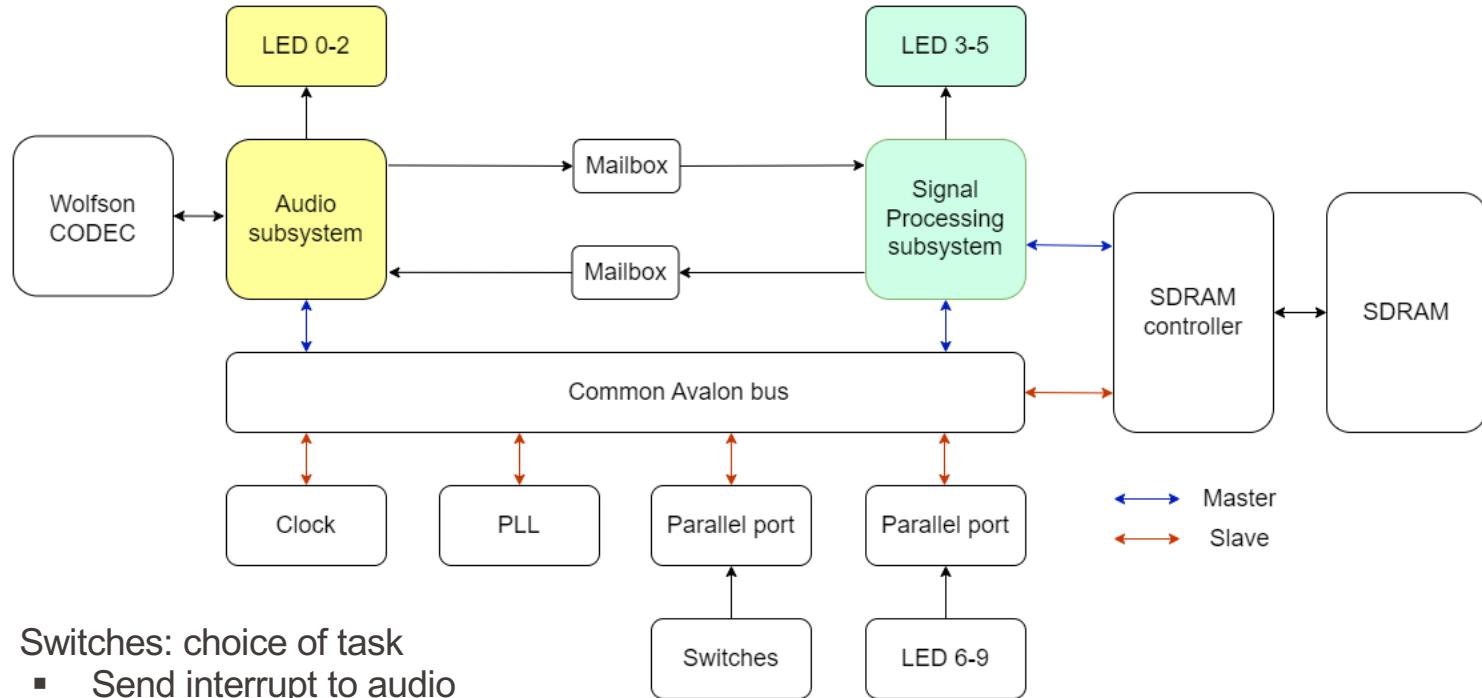
System functioning

Profiling

Conclusion

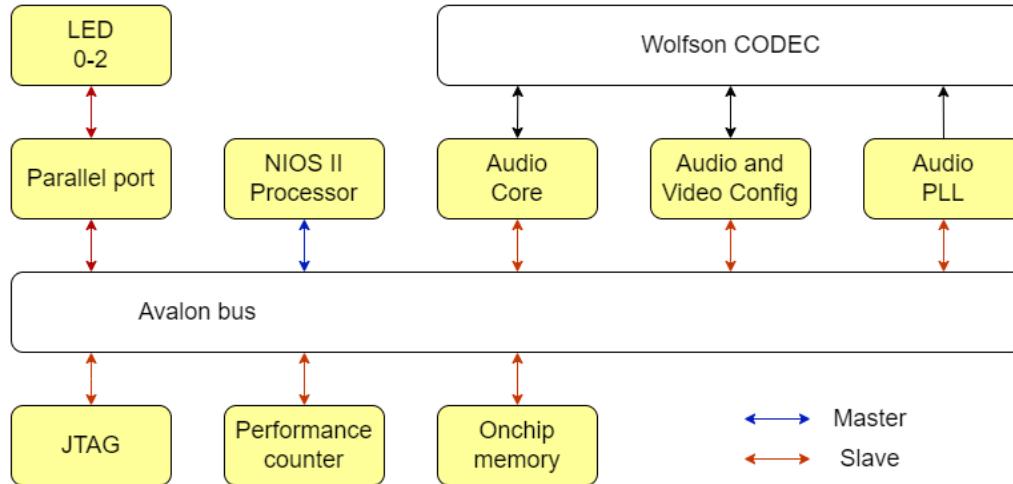
Demo

# System overview



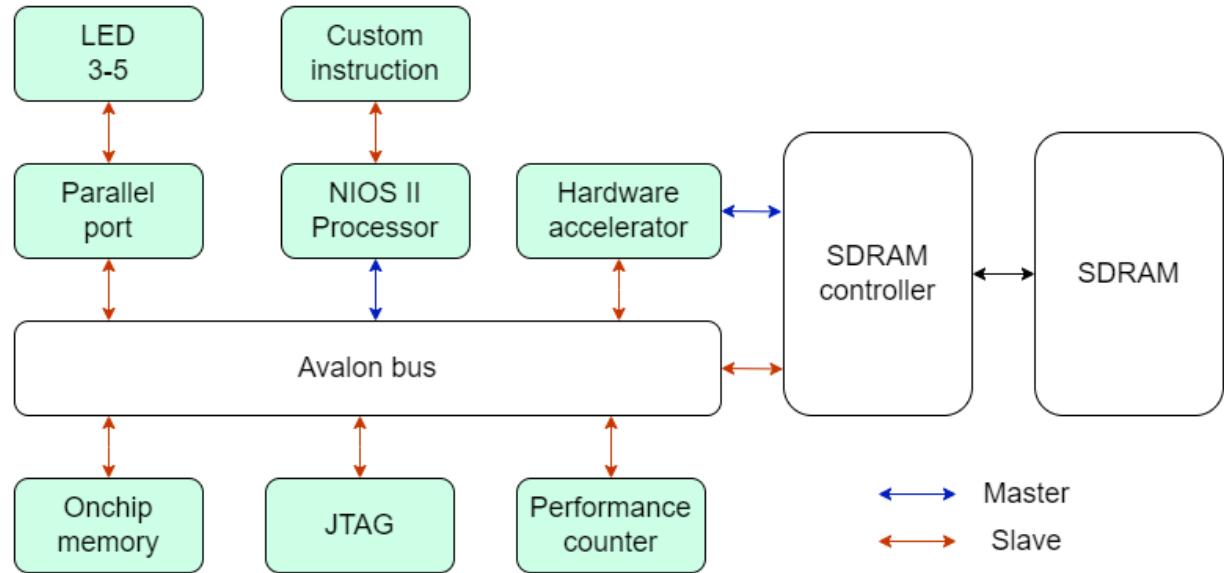
- Switches: choice of task
  - Send interrupt to audio subsystem
  - Audio subsys mails SP subsys
  - Still shared because of parameters

# Audio subsystem



- LED: early debugging
- Perf counter: profiling
- Onchip memory: program
- JTAG: printing in terminal
- Audio core
  - C interface
  - 2 FIFO pairs
- Config: setup
  - Fs: 48 kHz
  - ADC & DAC (32 bit digital)
- PLL: clock for CODEC
  - 12.288 MHz

# Signal processing subsystem



- LED: early debugging
- Perf counter: profiling
- Onchip memory: program
- JTAG: printing in terminal

# Custom instruction

- Changing volume = shifting data left/right
- Loading & storing of data in C: still slow!

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

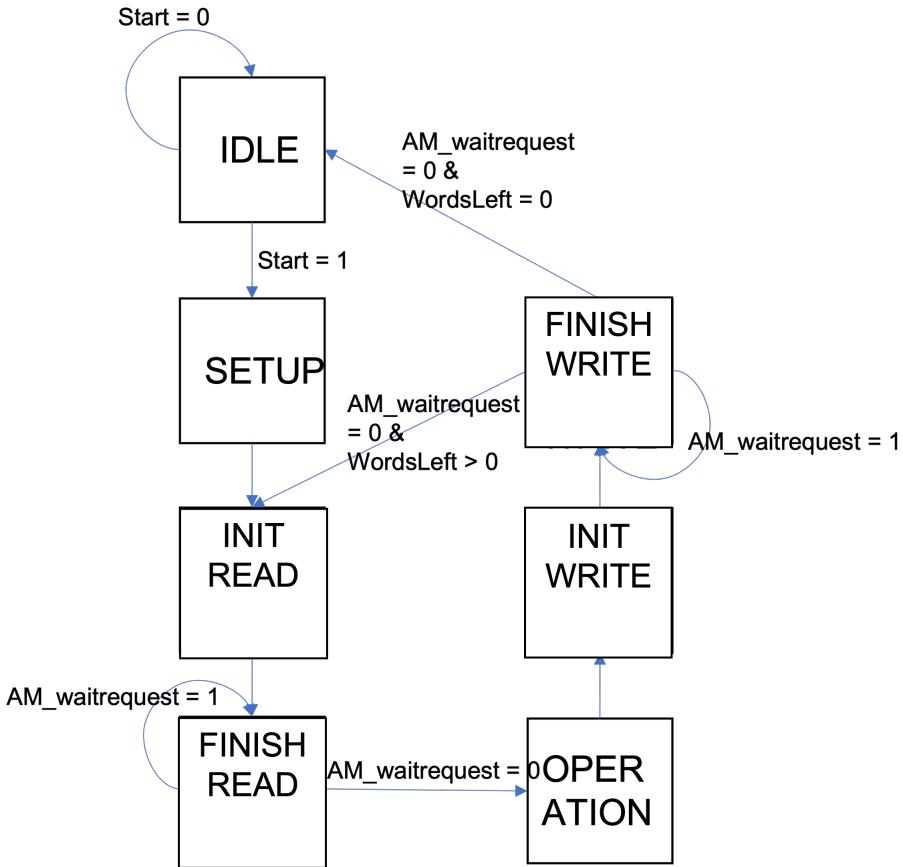
entity SigProcOperation is
  port(
    Param   : in std_logic_vector(31 downto 0);
    InData  : in std_logic_vector(31 downto 0);
    OutData : out std_logic_vector(31 downto 0)
  );
end SigProcOperation;

architecture design of SigProcOperation is
begin
  -- Choose volume up or down
  with Param(0) select OutData <=
    -- Increase the volume
    std_logic_vector(shift_left(signed(InData), 1)) when '1',
    -- Decrease the volume
    std_logic_vector(shift_right(signed(InData), 1)) when '0',
    InData when others;
end design;
```

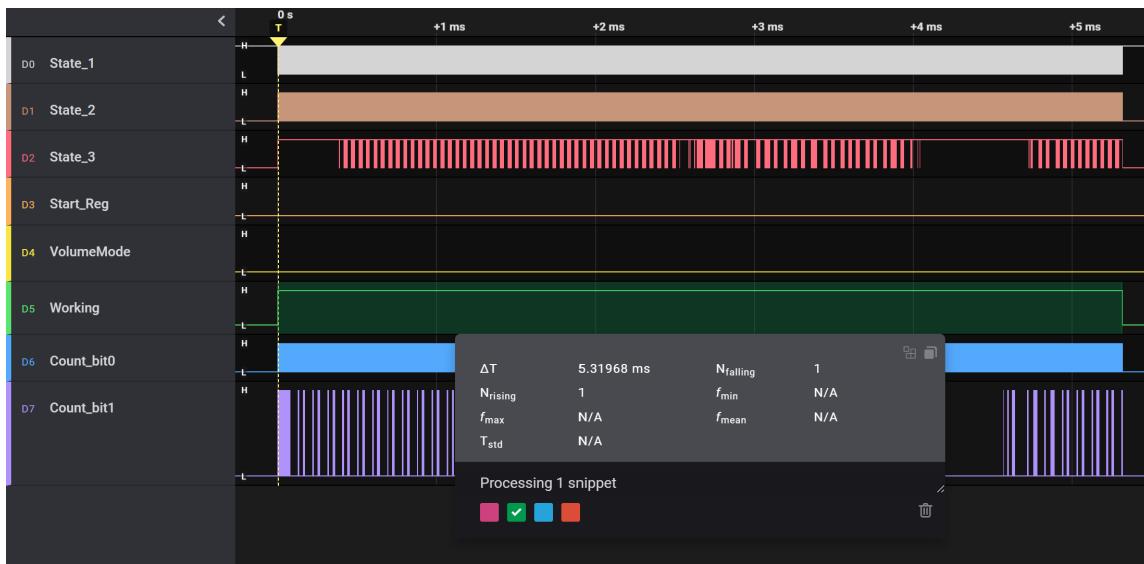
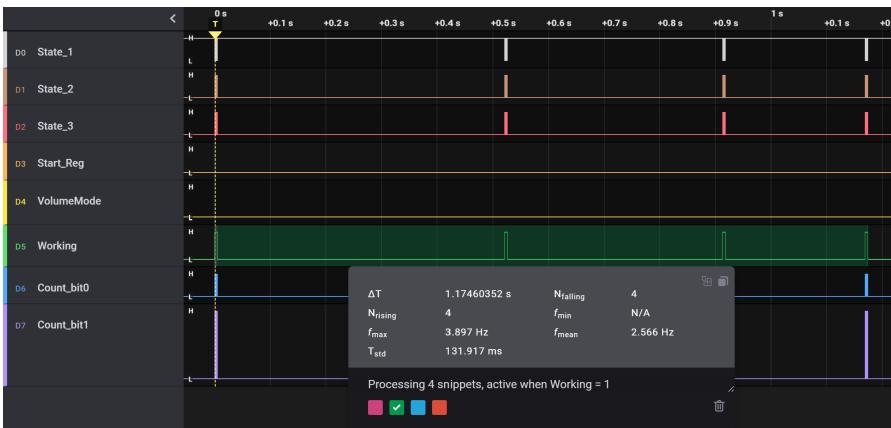
# Hardware accelerator

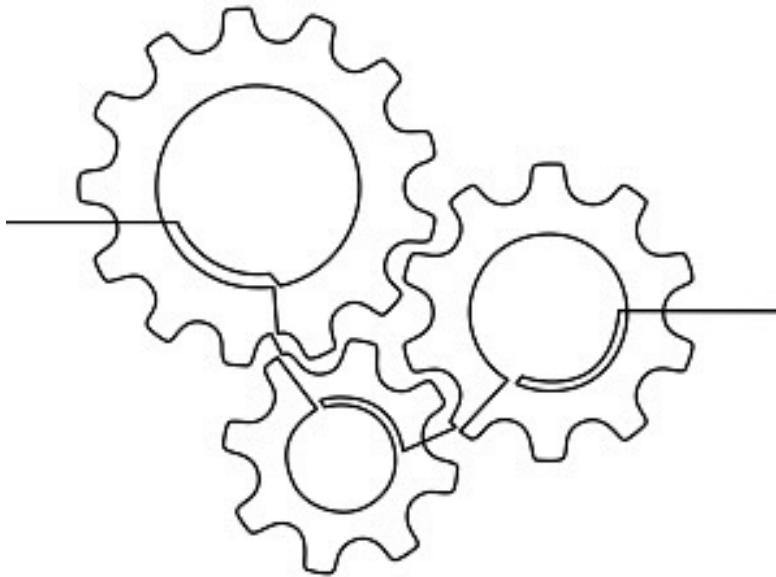
Main goal: freeing up processor

- DMA encapsulated in FSM



# Hardware accelerator debugging functionality





## OUTLINE

**Goal of the project**

**System description**

# **System functioning**

**Profiling**

**Conclusion**

**Demo**

# Switches

- SW0: record
  - SW1: play
  - SW7: volume up/down
- 
- SW2: profiling
  - SW3: verifying
  - SW4: audio device status
  - SW5: reset audio core
  - SW6: reset audio config

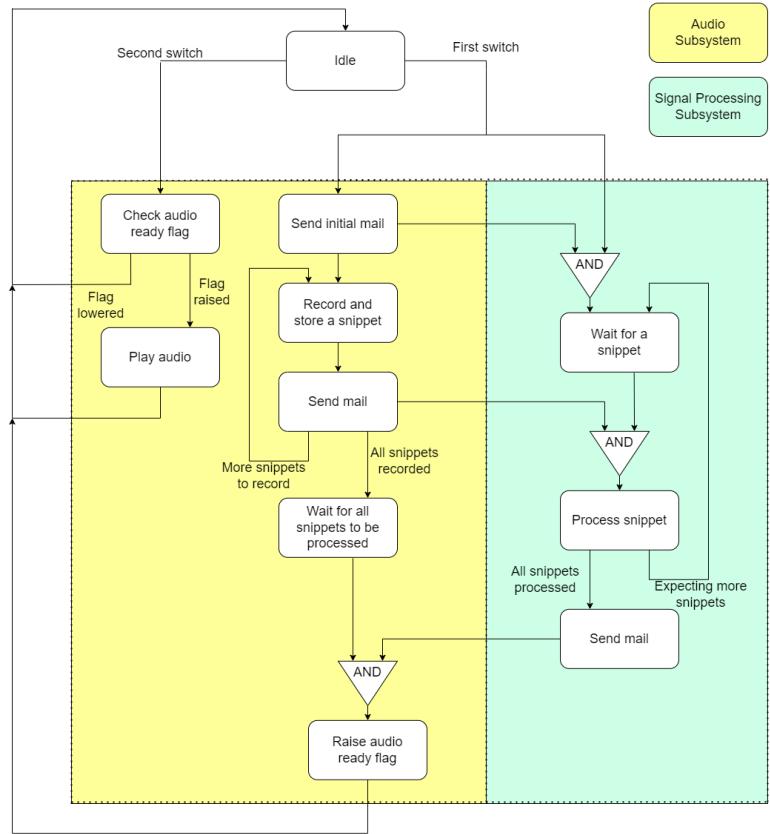


Recording & playback

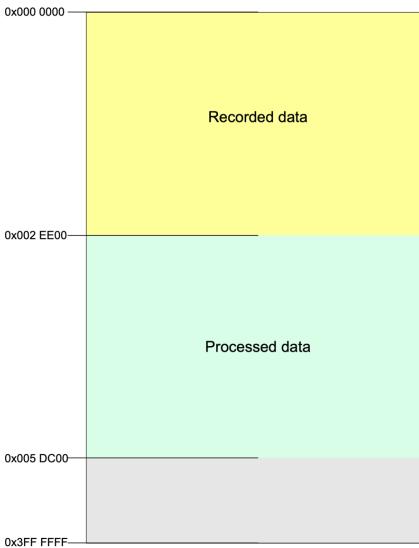


Debugging/troubleshooting & measuring

# Recording & playback workflow



- Only communication between subsystems by mail
- Audio divided into 12 000 sample snippets (0.25 seconds)
- Initial mail: what to expect (starting address of recording, amount of snippets)
- Every snippet: mail



# Debugging/troubleshooting & verifying workflow

```
Task 5: Checking audio device status...
Checking the settings of the Audio and Video Config device...
The status register is 30102
The Ready bit is 1
The Acknowledge bit is 0
Checking the settings of the Audio Core device...
The status register is 0
the FIFOspace registers are 7f808080 and 5a8c00
The ldada register is 5ed800
The rdata register is = 291500
Task complete. Ready for new task.
```

```
Task 6: Reset Audio Core...
Resetting Audio core...
Task complete. Ready for new task.
```

```
Task 7: Reset Audio and Video Config device...
Resetting Audio and Video Config device...
Successfully reset!
Task complete. Ready for new task.
```

```
Task 5: Checking audio device status...
Checking the settings of the Audio and Video Config device...
The status register is 30102
The Ready bit is 1
The Acknowledge bit is 0
Checking the settings of the Audio Core device...
The status register is 0
the FIFOspace registers are 7f808080 and 87fc00
The ldada register is 830f00
The rdata register is = 479f00
Task complete. Ready for new task.
```

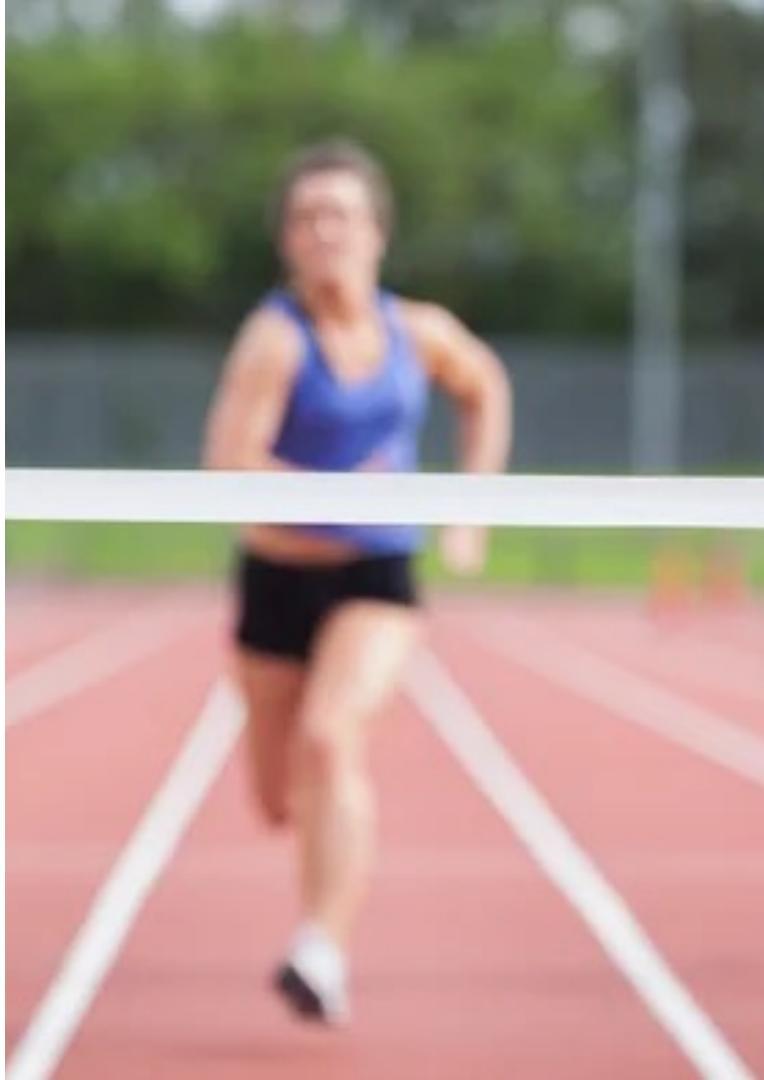
```
----- main -----
Task 1: Recording...
Sending initial mail, for a recording of 4 snippets...
Snippet recorded, sending mail...
Snippet recorded, sending mail...
Snippet recorded, sending mail...
Snippet recorded, sending mail...
Waiting for the Signal Processing subsystem to finish...
'Confirmation mail received.
Processed audio is stored at address = 2ee00, and of memory size = 2ee00
Task complete. Ready for new task.

Task 4: Verifying processed audio...
Test finished. Success rate 100.000000 %
Task complete. Ready for new task.
```

```
----- main -----
Task 1: Performing signal processing on the recording...
Initial mail received: Total snippets are 4
Operation type: Left shift chosen
Mail received: Snippet ready for processing
All snippets handled.
Sending confirmation mail with start_storage_address = 2ee00 and memory_size = 2ee00
Task complete. Ready for new task.

Task 4: Verifying processed audio...
Task complete. Ready for new task.
```

- Right: verification
- Left: Debugging
  - Only ldata & rdata change!



- Mini-project: Audio volume control

## OUTLINE

**Goal of the project**

**System description**

**System functioning**

**Profiling**

**Conclusion**

**Demo**

```
Task 3: Profiling...
Recording for HW accelerator...
Test finished. Success rate 100.000000 %
Recording for custom instruction...
Test finished. Success rate 100.000000 %
Recording for C code implementation...
Test finished. Success rate 100.000000 %
--Performance Counter Report--
Total Time: 3.32697 seconds (166348514 clock-cycles)
+-----+-----+-----+-----+
| Section | % | Time (sec) | Time (clocks)|Occurrences|
+-----+-----+-----+-----+
| Recording | 89.9| 2.99210| 149604854| 3|
+-----+-----+-----+-----+
| Verifying | 8.84| 0.29394| 14697045| 3|
+-----+-----+-----+-----+
| Waiting for #2 | 1.17| 0.03893| 1946653| 3|
+-----+-----+-----+-----+
Task complete. Ready for new task.
!
```

```
Task 3: Profiling...
Running on HW accelerator...
Running on custom instruction...
Running on c code implementation...
All profiling runs are finished.
--Performance Counter Report--
Total Time: 3.23009 seconds (161504663 clock-cycles)
+-----+-----+-----+-----+
| Section | % | Time (sec) | Time (clocks)|Occurrences|
+-----+-----+-----+-----+
| Accelerator | 0.659| 0.02128| 1064026| 4|
+-----+-----+-----+-----+
| Custom instr. | 2.08| 0.06716| 3358033| 4|
+-----+-----+-----+-----+
| C code | 2.05| 0.06625| 3312299| 4|
+-----+-----+-----+-----+
| Waiting for #1 | 95.1| 3.07241| 153620691| 15|
+-----+-----+-----+-----+
Task complete. Ready for new task.
```



■ Mini-project: Audio volume control

## OUTLINE

**Goal of the project**

**System description**

**System functioning**

**Profiling**

**Conclusion**

**Demo**

- Functionality implemented using dual core system
- Already start processing while recording for speed up
- Hardware accelerator for even more speed up



## OUTLINE

**Goal of the project**

**System description**

**System functioning**

**Profiling**

**Conclusion**

**Demo**

