

Lab 5 M2

ECEN 427

BYU Electrical & Computer
Engineering
IRA A. FULTON COLLEGE OF ENGINEERING

Lab 5

Goal: Play sound effects in space invaders

This chip converts digital samples to analog voltages

There is an I2S bus (Inter-Integrated Circuit Sound).
Digital samples need to be provided at 48kHz (20.83us).

Audio Controller

There is a digital circuit on the FPGA that has a 1024 sample queue. The queue data will be fed out at 48kHz.

How long does it take to empty the queue?

Not practical to keep filling the queue from user space.

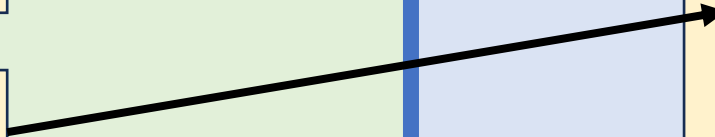
User Space

Kernel Space

Read WAV file
from filesystem



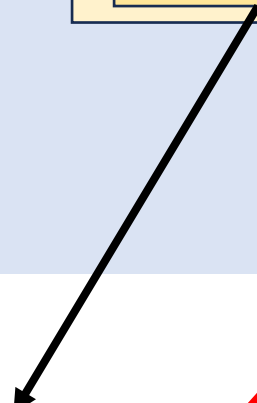
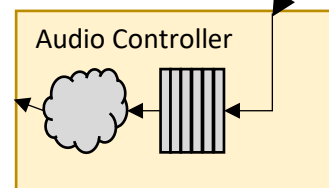
write()
WAV data



Kernel Driver

write
handler

Interrupt
handler



Interrupt
(When 75% empty)



- http://www-ug.eecg.toronto.edu/msl/nios_devices/datasheets/sound_codec_power_point.pdf

Some differences:

- Sample rate: Toronto says 44 kHz; ours is **48 kHz** (one sample every 20.83 μ s).
- Sample width: Toronto uses 16-bit; our codec takes **24-bit** samples in a 32-bit register.
- FIFO depth: Toronto's FIFO is unspecified; ours holds **1024** samples, and the interrupt fires when 256 or fewer remain.

Audio Documentation

- <https://byu-cpe.github.io/ecen427/documentation/audio-hw/>
 - Make sure to see info on provided WAVE file format

User Space

Kernel Space

Read WAV file
from filesystem



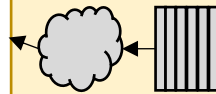
write()
WAV data

Kernel Driver

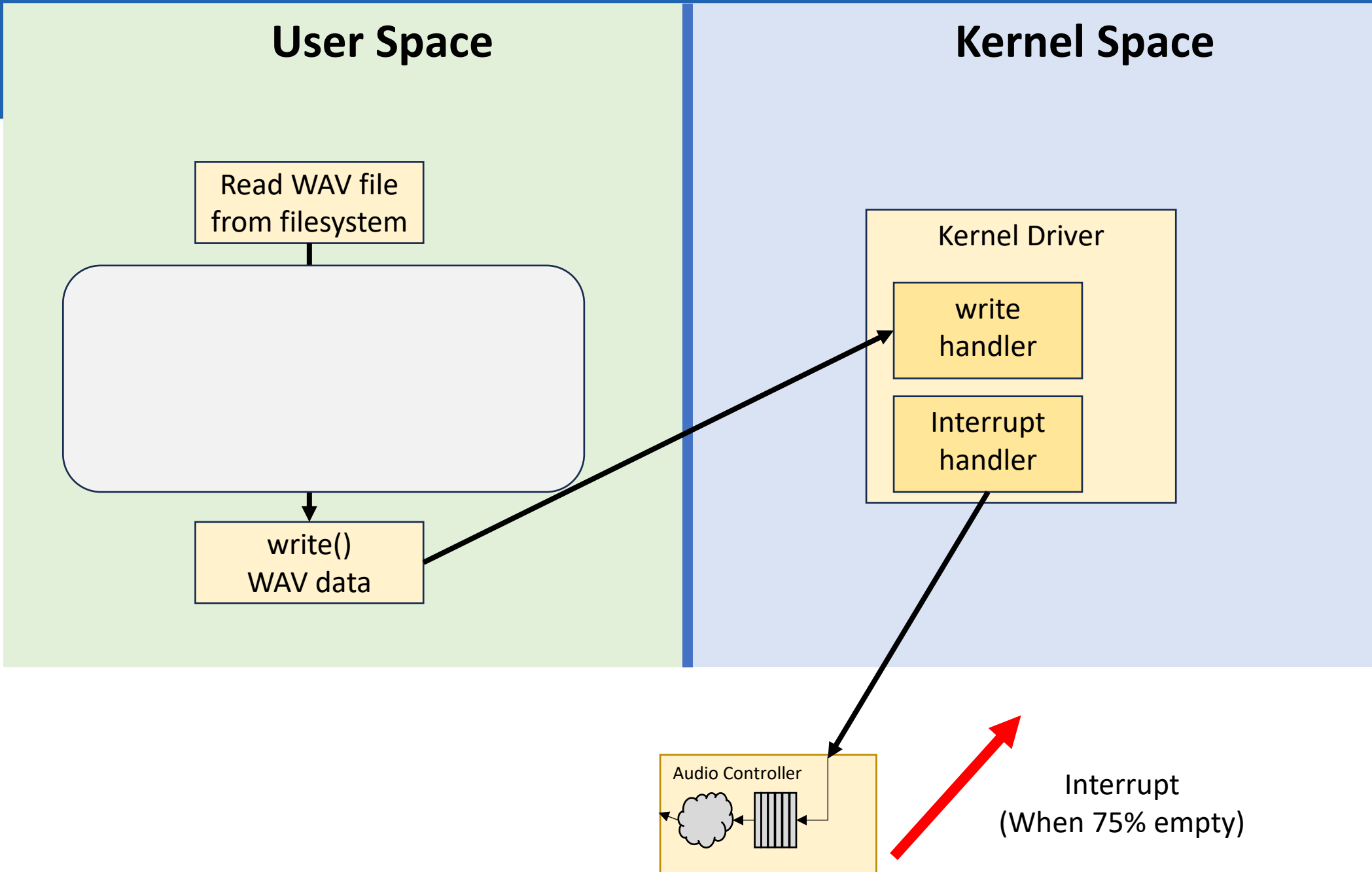
write
handler

Interrupt
handler

Audio Controller



Interrupt
(When 75% empty)



Inside write()

1. Disable the audio core's interrupts
2. Check that count fits in your buffer
3. Copy the data from user space: `copy_from_user()`
4. Reset / reinitialize:
 - sample index (where you are currently playing in the file)
 - the sample count (how many samples are in the current file)
 - a flag to track if you are currently playing
5. Enable interrupts
6. Return count

If you receive a new audio clip while the last one is still playing, immediately replace the current audio clip.

- If there are still audio samples to be played, play them
- Always put them in both LEFT and RIGHT FIFOs
- How many samples should you put in the FIFOs?
- Be careful as you reach the end of your audio clip
- Turn off interrupts once you reach the end of your audio clip

Inside read()

Recall how read is called from user space:

```
ssize_t read(int fd, void buf[count], size_t count);
```

Your driver should do the following:

- provide 1 byte back to the user, a 0 or 1, indicating if audio is currently playing
 - use `copy_to_user()`
- `read()` should always **return** a 1 (it always provides 1 byte back)
- The data written to the user's buffer will be a 0 or a 1

audio_driver_test2

- Call `audio_config_init()` first, then open `/dev/ecen427/audio_codec`
- Load the WAV file given on the command line
- `write()` the clip
- poll `read()` until it ~~returns~~ 0
- `write()` it again
- It must build to `userspace/build/apps/audio_driver_test2/audio_driver_test2`

Do not use sleep to wait.

Mixing up **samples** vs **bytes**

- What functions or code structures use number of samples?
- What functions or code structures use number of bytes?