

HW/SW Codesign LU - FLAC Decoder

Michael Platzter, Florian Zaruba, Thomas Weber

December 18, 2014

Table of contents

1 Partitioning

2 SDCard

3 FFT - Display

4 Software

Partitioning

Ideas for the Implementation

- SD-Card read in HW
- Display and FFT in HW
- Decode in Software
- Optimizations in C code (loop unrolling,..)

SDCard - Overview

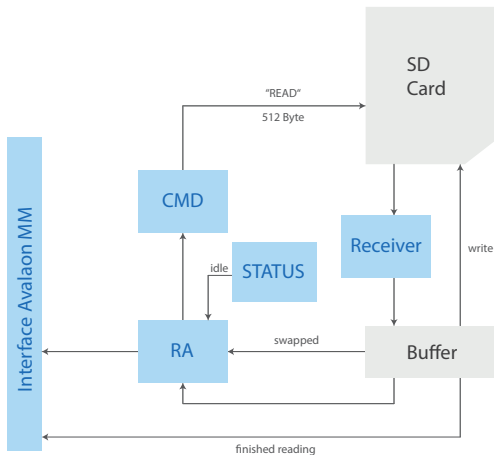


Figure : Read-ahead overview

RA - Buffer

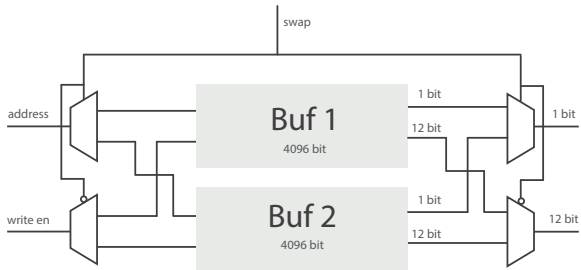


Figure : Read-ahead RTL

FFT-Display

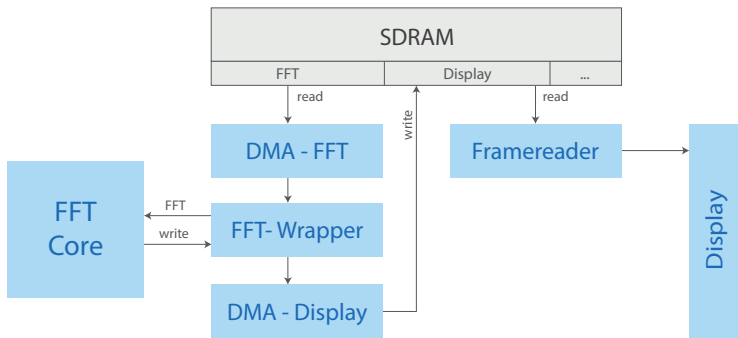


Figure : FFT Display overview

Bit Reader

- 512 bytes (4096 bits) read at once from SD card
- Stored in buffer with bytes of each 32 bit word inverted
- Bit reader context provides functions to read bits, which must not be aligned and can overlap words and blocks
- Functions match requirements to decode FLAC frames

```
int get_bits(int n);      // read up to 32 bits
int get_unary1(int max);  // read unary encoded value
...
```

Software - Get Bits

```
uint32_t buf[128]; // 512 byte = 4096 bit buffer
int pos;           // bit position in buf

unsigned int get_bits(int n)
{
    if (pos >= 4096)
        read_block(); // read next sd card block into buf

    // read first word into cache:
    uint64_t cache = (uint64_t)buf[pos >> 5] << 32;

    // if bits to be read overlap two words, read next word:
    if ((pos & 0x1f) + n > 32)
        cache |= (uint64_t)buf[(pos >> 5) + 1];

    // shift out bits which must not be returned:
    uint64_t ret = (cache << (pos & 0x1f)) >> (64 - n);
    pos += n;
    return ret;
}
```


Software - LPC decoding

Decode LPC coded samples:

s_i : sample, r_i : residual, c_i : LPC coefficient, p : predictor order

$$s_n = r_n + \sum_{i=0}^p s_{n-i} \cdot c_i$$

Naive code with nested loops:

```
for (i = pred_order; i < len; i++, decoded++) {  
    int sum = 0;  
    for (j = 0; j < pred_order; j++)  
        sum += coeffs[j] * decoded[j];  
  
    decoded[pred_order] += sum >> shift;  
}
```

Optimize to reduce branches and memory accesses