

roadmap

☐ 实现数据传输

☐ 内存内容 → BPSK

■ 模块

☒ `adc_axis`

1. 从 ADC 取数据到 AXIS DATA FIFO
2. 当 FIFO 将满时停止取数据，并将事件发送到 Zynq 外部中断

☒ `axis_data_fifo_adc`

1. 暂时储存 ADC 数据

☒ `axi_dma_0`

1. 在系统总线上搬运数据
 - `axis_data_fifo_adc` ==> DDR
 - DDR ==> `axis_data_fifo_dac`

☒ `dac_axis_0`

1. 不间断从 `axis_data_fifo_dac` 取数据
2. 无数据时保持 0V
3. 取数据间隔 div
4. `axis_data_fifo_dac` 为空时 busy

☒ `mux_data_available`

- 当 `dac_axis_0` 不忙时输出到 DAC

☒ `DUC`

1. 暂时接入静态时钟
2. 将 FIFO 最低位作数据位，进行 BPSK 编码

☒ `mux_reciever_out`

1. 控制数据输出类型
2. 类型由 Zynq 指定

■ 方式

☒ 连续式

☐ 中断式

1. Zynq 速度可能不够，中断触发速度较慢

☐ BPSK → 内存内容

■ 模块

☒ `DDC`

1. 暂时接入静态时钟
2. 将 ADC 输入的 BPSK 波形转换为单 Bit
3. 暂未实现能量触发

☒ `mux_reciever_in`

1. 控制数据输入类型
2. 类型由 Zynq 指定

■ 方式

☒ 连续式

☐ 中断式

■ TODO

■ 能量触发

☐ 系统

☐ Zynq

- ☐ 访问有线网络

☒ GPIO 输入

☒ GPIO 输出

☒ 外部中断触发
1. ADC FIFO prog_full: ADC FIFO 容量大于某值

2. DMA mm2s: MEM => Device done

3. DMA s2mm: Device => MEM done

4. DAC FIFO almost_full: DAC FIFO 容量将满

5. DAC FIFO almost_empty: DAC FIFO 容量将空

6. ADC FIFO almost_full: ADC FIFO 容量将满

7. ADC FIFO almost_empty: ADC FIFO 容量将空
- ☐ 调试

☒ SystemILA

☒ SLOT0: ADC_FIFO → DMA

☒ SLOT1: DMA → DAC_FIFO

☒ SLOT2: ADC_AXIS → ADC_FIFO

☒ SLOT3: DAC_FIFO → DAC_AXIS

☒ 其他状态

☒ DEBUG_CTRL : GPIO-IN & OUT

1. DUC_SYNC : 是否使能 DUC

2. CLK_PSEN : 动态时钟偏移

3. CLK_PSCLK : 动态时钟偏移

4. CLK_PSINCDEC : 动态时钟偏移

5. RECIEVER_OUT : 输出 DUC 还是原数据

6. DDC_SYNC : 是否使能 DDC

7. RECIEVER_IN : 输入 DDC 还是原数据

8. DAC : dac_axis.div 模块输出分频

9. ADC : adc_axis.div 模块输入分频

10. ADC_FIFO_RESET : ADC FIFO 手动复位

11. DAC_FIFO_RESET : DAC FIFO 手动复位

☐ 当前设置细节

1. adc_axis 在 adc_fifo.prog_full 时将会发送 tlast 并停止直到 almost_empty , 等待 DMA 读取完毕

2. fixed: tlast 和 tready&&tvalid 对齐

3. prog_full.threshold 设置为 2048

4. 设置 DMA 后设置 SetupDebugBits() , 即设置 adc_axis.div > 0 即开始写入 FIFO

5. Zynq 处理 DMA 读取到的数据的时间较长, 暂未实现连续中断触发, 所以将单次处理 Bytes 数量减小

6. 两个 FIFO 当前大小为 16384 Bytes, 使用 500k 频率的 ADC 可以在 3.2ms 内填满 FIFO

7. DMA 读取到的数据数量是不确定的, 由 FIFO 实际含有的数据量、写入时 tlast 状态决定

8. 下列情况下 DMA 读取会卡死或无数据

1. tlast 与 tvalid & tready 不对齐。ADC FIFO 的 tvalid 有时候不太准确, 没有空间写入的时候 tvalid 的拉低可能会延迟 1~2 周期, 造成 tlast 拉高情况未写入而造成 DMA 未读取到数据尾

2. 一直拉高 tlast , 可能会被判断为上次传输未完成, 而不是本次传输只传输 1 次...

3. 提供的 len 大于寻址地址线寻址范围, 本来使用 14 位地址线而读取 16384 字节数据 (2¹⁴) 会出现这种情况

☐ 实现网络数据包传输

☐ 实现 P2P 网络
- todolist

- ☐ 设置能量触发电路
- ☐ 时间测量
 - ☐ 测量无线部分延迟
 - ☐ 测量从系统发出包到接收到包的时间
- ☐ 读取数据清理
 - 1. 前后可能多出几 Bit
 - 2. 如何校验比特位置
- ☐ 设置为动态时钟
 - ☐ 可能造成电路不稳定，需要使用跨时钟 FIFO
- ☐ 将数据传输打包为数据链路层网络包