SPI Driver Design with DMA (Embedded Systems)

Goals & Constraints

- Use DMA to offload SPI transfers
- Support full-duplex
- Blocking and non-blocking APIs
- Handle CS, cache coherence, errors

Architecture

Application -> SPI Driver -> SPI peripheral + DMA (TX/RX) + CS control
Streaming: Device -> SPI -> DMA (circular) -> Ring buffer -> Consumer task

Public API

spi_transfer_sync(h, tx, rx, len, timeout)
spi_transfer_async(h, tx, rx, len, cb, ctx)
spi_start_stream_rx(h, buf, size)
spi_stop_stream_rx(h)

Data Structures

spi_t { state, mutex, sem, cb, cs_port/pin, dma_tx, dma_rx, rx_ring, rx_head/tail }

DMA Flow

1. Configure RX DMA
2. Configure TX DMA (dummy if needed)
3. Assert CS
4. Start RX then TX
5. Wait (blocking) or return (async)
6. On DMA complete: deassert CS, cleanup

Cache & Alignment

- Clean TX buffer before DMA
- Invalidate RX buffer after DMA
- Align to cache line size if needed

Chip Select

- CS low before transfer, deassert after complete
- Respect device timing requirements

Timeout & Errors

- Abort DMA if timeout
- Handle SPI overrun/CRC errors

Interrupts

- Use DMA complete IRQs (RX preferred)
- ISR signals semaphores or schedules callback

Streaming

- Use circular DMA RX into ring buffer
- Half/full-transfer interrupts -> notify consumer

Concurrency

- Serialize access with mutex
- Optional transfer queue

Generic C Driver Template

```
spi_status_t spi_transfer_sync(spi_t *h, const uint8_t *tx, uint8_t *rx, size_t len, uint32_t timeout_ms) {
    dma_setup_rx(...);
    dma_setup_tx(...);
    gpio_cs_assert(...);
    dma_start(rx);
    dma_start(tx);
    wait_on_semaphore(...);
    gpio_cs_deassert(...);
    return SPI_OK;
}
```

STM32 HAL Example

```
HAL_SPI_TransmitReceive_DMA(&hspi1, txBuf, rxBuf, len);
void HAL_SPI_TxRxCpltCallback(SPI_HandleTypeDef *hspi) {
    gpio_write(CS, 1);
    xSemaphoreGiveFromISR(...);
}
```

Testing Checklist

- 1-byte transfer
- Large transfer
- Full-duplex test
- Timeout handling
- Cache coherency
- Multi-client test
- Streaming wraparound

Common Pitfalls

- Must start RX before TX
- Cache bugs on Cortex-M7/A
- DMA alignment restrictions
- FIFO/errata issues