ChipTune

- Author: Wallie Everest
- Description: Vintage 8-bit sound generator
- Language: Verilog

How it works

Chiptune implements an 8-bit Programmable Sound Generator. Input is from a serial UART interface. Output is 4-bit DAC audio.

The ChipTune Project: This is a two-in-one project. First, an audio device replicates the square-wave sound generators of vintage video games. Second, a re-imagined version of the scan controller is implemented to demonstrate faster data acquisition.

TinyTapeout 3 Configuration: For this third Multi Project Chip (MPC) tapeout, the original scan chain will be configured in 'external mode'. The inputs and outputs for user projects will be derived externally from scanchain data, not from the chip's I/O pins. The scanchain control signals will occupy pins designated io_in and io_out. Expected throughput is better than 10k byte/second. The ChipTune project will configure the shift clock to attain 9600 bytes/sec. This speed provides a 4800 Hz clock and 300 baud communication to the tiny user project. Devices from the eFabless Multi-Project Wafer (MPW) shuttle are delivered in two package options, each with 64 pins. TinyTapeout 2 will be packaged in a QFN, whereas TinyTapout 3 will be packaged as a WCSP. In both delivered configurations, the device is mounted to a daughter board PCB with 10 castellated pins per side. The daughter board provides basic clock, configuration and power management for the Caravel SoC.

Caravel Connections: Within the Caravel SoC, the TinyTapeout project has configured the user space into 250 sub-projects. Each project is interconnected by a scanchain that serpentines through the chip. Control of the 4-wire chain provides access to each project. The scanchain topology has pros and cons, as would any interconnect scheme. This project presents an alternative topology based on a JTAG implementation. The advantage is a reduction in the length of the register latency.

Scanchain V1 - Pro: Economical use of resources. Readily hardened in ASIC fabric. Testable on an FPGA platform. - Con: Very long register chain (2,000) impacts overall acquisition rate.

Scanchain V2 - Pro: Short register chain (10) minimizes latency. Economical use of resources. Testable on an FPGA platform. - Con: Speed limited by length of multiplexer propagation (250 instances).

Multiplexer - Pro: Pure combinatorial output. Potential to be the fastest option. - Con: Requires a large number of long routing resources. Internal tri-state busses not testable on an FPGA platform.

Tiny Project Configuration: This tiny project embeds the revised scanchain (version 2) to investigate its low latency and testability. The penalty for this implementation is 10 clock cycles per transfer. The delay is mitigated by a serial UART expanding the internal registers of the audio generator. Four extra scanchain endpoints are included for demonstration purposes. A clock generator is used to recover a bit-clock from asynchronous serial data. An external 16x baud clock is required by the design. This planned 4800 Hz project clock will permit a 300 baud serial link.

Scanchain Version 2: The re-imagined scanchain uses a bypass technique commonly seen with JTAG devices. Each 'tap' in the chain routes data through a combinatorial multiplexer until the module is activated. Individual taps are assigned a unique 8-bit address during tapeout elaboration. A tap is activated when it receives a matching address message, enabling its' 8-bit shift register. Unselected taps drive logic 0 into the projects' inputs to keep them in a quiescent state. Encoding of serial data is compatible with the ubiquitous UART format. The waveform is one-start, eight-data, one-stop (300,8,n,1). Least significant bits are transmitted fist. An immediate advantage is the use of a computer COM port to generate and analyze functional data. The serial interface is in addition to decoded parallel data available on the I/O ports.

ChipTune Operation: The audio portion of the project consists of two rectangular pulse generators. Each module is controlled by four 8-bit registers. Configurable parameters are the frequency, duty cycle, sweep, decay, and note duration. The frequency range of the project is limited by the legacy scanchain, but mid-range frequencies are acceptable. Additional triangle and noise modules will be added in future work when more bandwidth is available.

Design For Test Considerations: An isolated instance of scanchain version 2 has been tested on an FPGA platform with good results. A shift rate of 3.7 MHz permits communication with the computer at 115,200 baud. Longer scan chains do not affect throughput until multiplexer delays become dominant. For an FPGA, 75% of the timing delays are attributed to routing resources. Output of the sub-project is always available at both the parallel output port and the serial data. A 'Mode' signal driven by the DTS line controls whether the project's input is derived from the parallel input port or serial data.

Summary: The next shuttle for TinyTapeout is planning a multiplexer for selecting between the 250 projects. This will alleviate latency in the present design. The revised scanchain offered here is an alternative for other group projects. The scanchain topology still holds merit in many applications. The application of a computer-based logic analyzer via a USB COM port is appealing.

How to test

The ChipTune project can be interfaced to a computer COM port at 300 baud. The project's scanchain is updated at 9600 Hz such that a 4800 Hz clock is available at io_in[4]. An external bit clock must be provided on io_in[5] with a rising edge in the center of the bit period. This clock may be source from the reference clock on io_out[4]. Serial data is received on io_in[7]. Serial data is decoded by the device as an address when io_in[6] = 0, and as data when io_in[6] = 1. A 4-bit DAC output is available on io_out[3:0].

IO

#	Input	Output
0	io_in[0] (CLOCK)	io_out[0] (DAC_0)
1	io_in[1] (IN_1)	io_out[1] (DAC_1)
2	io_in[2] (IN_2)	io_out[2] (DAC_2)
3	io_in[3] (MODE)	io_out[3] (DAC_3)
4	io_in[4] (BAUD_CLK)	io_out[4] (REF_CLK)
5	io_in[5] (TCK)	io_out[5] (RTCK)
6	io_in[6] (TMS)	io_out[6] (LED)
7	io_in[7] (TDI)	io_out[7] (TDO)