Benny Hui

236-518-2518 • Vancouver, BC • 2005bennyhui@gmail.com • bennyhui.net • GitHub • LinkedIn

SKILLS

Programming Languages: Python, C, C++, 8051 Assembly, AArch64 Assembly, HTML/CSS, Arduino, SQL
Systems & Hardware: FPGA, System Verilog, EFM8, STM32, N76E003, ATMega328P, DE1-SoC, Soldering, Oscilloscope, Multimeter, AXAU15, Te0-712
Technologies & Tools: Git, MATLAB, Quartus, ModelSim, MS Office, STM32CubeIDE, SolidWorks, Altium Designer Picoblaze, PuTTY, DOSBox, AMD Xilinx Vivado, SVN Tortoise

EDUCATION

University of British Columbia | Bachelor of Applied Science in Electrical Engineering (3rd Year) Expected May. 2028

- Co-op: available up to 4 months starting May 2026
- Courses: Circuit analysis in AC and DC with Electronics, Power Systems, Computer Architecture, Microcomputers, Data Structures & Algorithms, Signals & Systems, Electromagnetics (Magnetostatics & Electrostatics)

TECHNICAL EXPERIENCE

Bosch Sept. 2025 – Present
Digital Design Engineer Co-op | Reutlingen, Germany

- Pre-Development for next generation MEMS inertial measurement unit for automated driving
  AMD Xilinx Vivado | Integrating module blocks together which includes usage of IP cores for **Asynchronous FIFO** to transmit and receive data from alternate clock domains (Due to FTDI chip transmitting or receiving data to FPGA). Usage of **MIG DDR3** to transfer data into DDR3 memory. Test and debug modules via simulation.
- Used the Te0-712 (Artix 7 Family) and developed user logic as an FSM to allow for proper writing of data into DDR3 memory, along with allowing reading of data from memory at said address with IP cores that include **Clocking Wizard** with input clocks as **system clock differential** (sys_clk_p/n) and **MIG DDR3 SDRAM**
- Debugged using AMD Xilinx Vivado Simulation tool, IP core **ILA/"Setup Debug"** and creating testbench for FSM

TECHNICAL PROJECTS

FPGA Audio Player May 2025
System Verilog, Quartus, ModelSim, DE1-SoC | **Google Drive**

- Worked with **flash memory** extracting audio data from .jic file, using an **FSM** to have states of execution of collecting the data from said address
- Used **SignalTap** to visually inspect the specific audio data at said address along with the specific state within the **FSM**
- Used PS2 keyboard as user interface to start, stop, restart, play forward and backwards which act as **flags** for the **FSM**
- Debugged using **ModelSim** with **test bench** to determine states and view **waveform** to expect said execution/operation
- Wrote **test bench** to determine functionality of data path given several different example inputs
- Created **clock divider** to allow for extraction of audio data at alternate clock frequency allowing to speed or slow paste of song being played based off default clock frequency of De1-SoC (50MHz)

Coin Picking Robot Mar. 2025 – Apr 2025
C, Makefiles, ATMega328P, EFM8 | **GitHub** , **YouTube**

- Developed a remote-controlled robot with **master** (ATMega328P) **slave** (EFM8) configuration using JDY40 radio communication to send input signals to slave and adjust controls from user inputs
- Controlled servo motors using **PWM** with **timers** to change the direction of the robot without halting the process of detecting coins
- Debugged controller and robot through **UART**, communicating over TX/RX lines, displayed onto **PuTTY** allowing to determine working ranges and fail conditions
- Configured hardware on robot with **optocoupler** for robotic arm and **H-bridge** for **DC motors** to control robot wheels

Reflow Oven Feb. 2025 – Mar 2025
Python, 8051 Assmebly | **GitHub** , **YouTube**

- Deployed a **finite state machine** (FSM) to imitate the process of a reflow oven.
- Used **timers** to increment the timing of the reflow process at set state sending **PWM** signals to an SSR box which changes the temperature based on the current state of the **FSM**
- Used **ADC** button configuration to MCU to change reflow time, reflow temperature, soak time, and soak temperature
- Used **Python** plotting real time graph of temperature to time throughout the reflow process