

Progetto Logic Hacker

1 – Componenti del gruppo

1. Jacopo Belli
2. Giulio Fieramosca
3. Luca Mattii

2 – Obiettivo del progetto

Il progetto nasce dalla volontà di riunire e migliorare le funzioni del Bus Pirate e dell'Open Bench Logic Sniffer, entrambi prodotti open source del gruppo Dangerous Prototypes. Il Bus Pirate nasce come uno strumento open source per la prototipazione rapida basato su microcontrollore, in grado di fornire un'interfaccia universale tramite terminale PC per comunicare con periferiche che supportano un protocollo standard digitale. L'Open Bench Logic Sniffer, invece, è un analizzatore di stati logici open source realizzato con un FPGA che acquisisce i dati e un microcontrollore che, interfacciandosi con un PC, trasferisce la lettura dei segnali ad un client Java che la presenta graficamente all'utente.

Obiettivo del progetto è unire le funzionalità dei due oggetti sotto un unico sistema, basato su FPGA e soft-core Nios II, sviluppando opportune periferiche in linguaggio Verilog per supportare le operazioni richieste. In particolare, il Logic Hacker sarà in grado di leggere i dati presenti su un bus digitale, interpretarli in caso si stia osservando un protocollo standard e comunicare tramite gli stessi protocolli con periferiche esterne. Basare il progetto su un FPGA garantisce versatilità massima per il sistema, il quale sarà in grado di eseguire analisi di stati logici di nodi sia interni al chip, includendo in fase di testing il design a fianco di una piattaforma da analizzare, che esterni, semplicemente collegando sui pin esterni del chip gli ingressi logici dell'analizzatore.

Le milestone da raggiungere durante il progetto sono le seguenti:

1. Progettare il modulo Verilog di una porta parallela a 32 bit dotata di interrupt e maschera di trigger per l'avvio del campionamento. Il modulo, comunicando con Nios, fornirà fino a 4096 campioni del bus cadenzati a una frequenza impostabile da terminale. Le impostazioni del campionamento sono impostabili tramite il terminale. Il microprocessore dopo il prelievo dei dati provvederà a indirizzarli al computer tramite una porta seriale, sfruttando il protocollo standard che è richiesto dall'applicazione Java per graficare i segnali.
2. Scrivere il firmware per Nios II necessario alla decodifica dei protocolli standard UART, SPI, I2C.
3. Progettare il modulo Verilog di una periferica UART Master/Slave e del firmware necessario a configurarla tramite terminale.
4. Progettare il modulo Verilog di una periferica SPI Master/Slave e del firmware necessario a configurarla tramite terminale.
5. Progettare il modulo Verilog di una periferica I2C Master/Slave e del firmware necessario a configurarla tramite terminale.