

# Einstieg in Digilent's NetFPGA SUME

**NetFPGA SUME - ein Xilinx Virtex 7 FPGA-Board für Hochleistungsrechnen und Vernetzungssysteme**

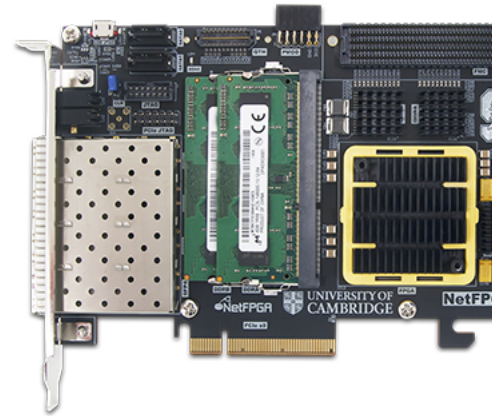
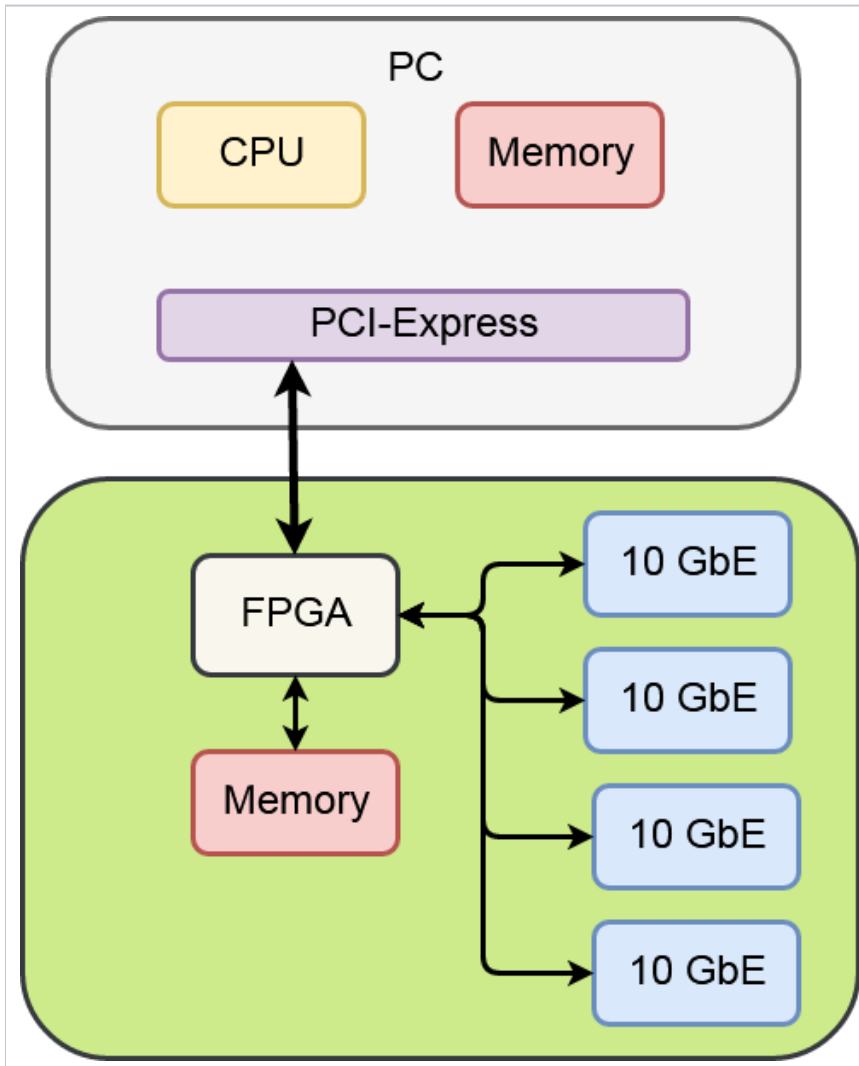
- [Einführung](#)
  - [Systemanforderungen](#)
  - [Infrastruktur](#)
  - [Hardware-Test](#)
  - [Referenzdesigns](#)
    - [Erste Referenz Schaltprojekt](#)
  - [Referenzen](#)
- 

## Einführung

Das NetFPGA-Projekt ist eine Gruppe, die Open-Source-Hardware und Software für schnelle Prototypenentwicklung von Hochgeschwindigkeits-, Hardware-beschleunigten Vernetzungssystemen entwickelt. Das NetFPGA-Projekt wird von der Field Programmable Gate Array (FPGA)-Technologie ermöglicht, so dass die Benutzer Pakete auf Zeilenrate bearbeiten können.

Das NetFPGA-SUME-Board ist für Hochleistungsrechnen und Vernetzungsdesign hoher Dichte geeignet. Es wird von by Xilinx Virtex-7 690T aktiviert und wird von Digilent, Xilinx, der Universität Cambridge und Universität Stanford mitentwickelt. Das Board wurde in akademischen und industriellen Forschungen inkl. Netzsicherheit, Software definierte Vernetzung, Hochleistungsnetzsysteme benutzt.

Dieses Einführungshandbuch deckt Systemanforderungen, Infrastruktur, Hardwaretest, Referenzdesigns und erste Referenz Schaltprojekt.



## Systemanforderungen

### 1. Betriebssysteme

- Sie können alle Betriebssysteme, die [Xilinx Vivado Design Suite](#) unterstützt, benutzen.
- Das NetFPGA-Team entwickelt strikt auf [Linux](#), also die Software-Bestandteile werden für components [Linux](#) entwickelt. Wir empfehlen den Benutzern über [Ubuntu 16.04](#) zu verfügen.

### 2. CAD-Instrumente / IDE

- [Xilinx Vivado Design Suite](#) wird verlangt. Die empfohlene Version ist Xilinx Vivado 2016.4.
- Eine gültige Lizenz für Xilinx 10G MAC.

### 3. Netzwerkkabel

Das NetFPGA-SUME-Board hat vier SFP+ Anschlüsse für 10G Ethernet. AVAGO AFBR-709SMZ Empfänger mit Multimode OM3 10Gb Fiber wird empfohlen

### 4. Host-Mainboard:

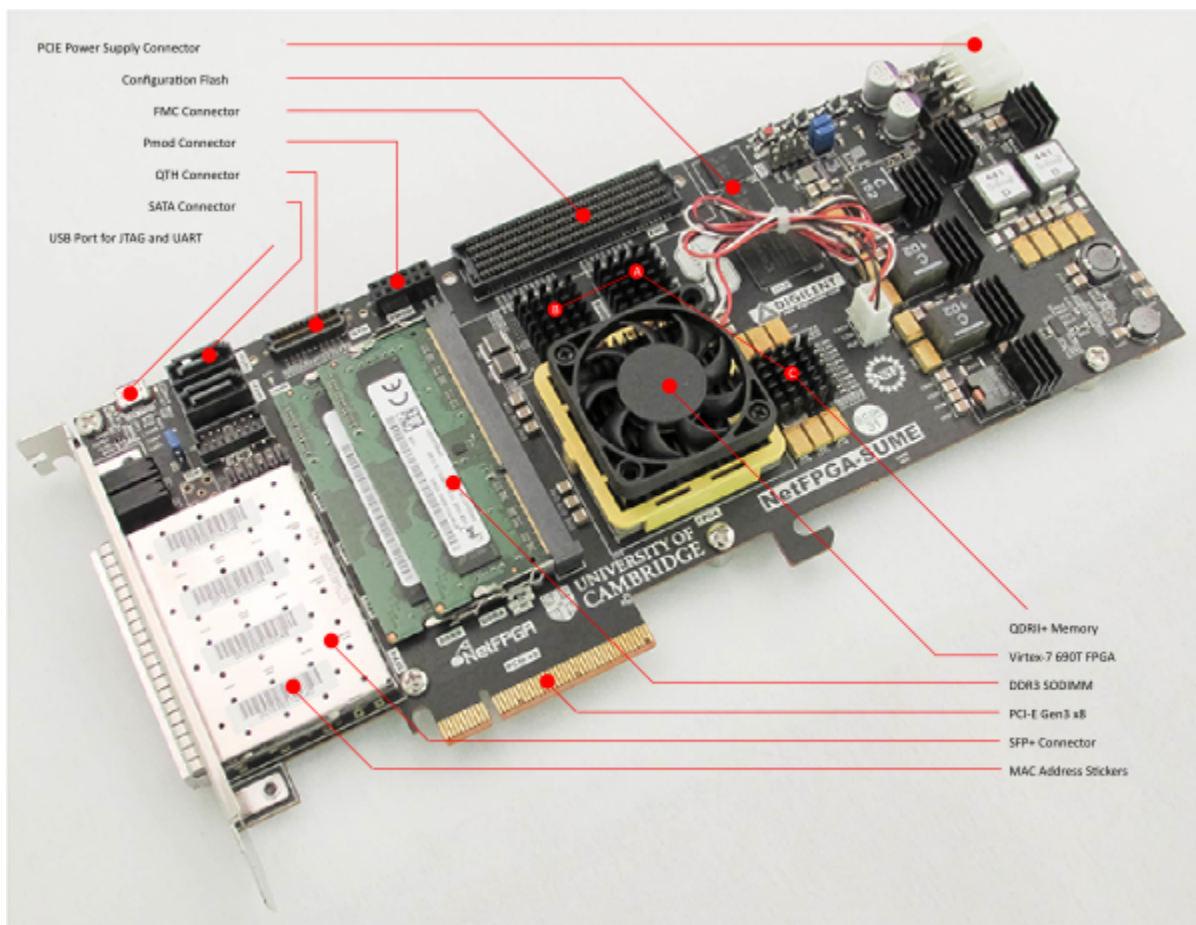
Sie können das NetFPGA-SUME-Board eigenständig und in einem Host benutzen.

- a. Eigenständig: Sie brauchen ein mikro-USB-Kabel für JTAG-Kette und serielle Kommunikation.
- b. Host: Das Mainboard muss PCI Express Gen3 x8 unterstützen. Ausführliche Informationen finden Sie unter <https://github.com/NetFPGA/NetFPGA-SUME-public/wiki/Motherboard-Information>.

## Infrastruktur

Das NetFPGA-SUME-Set beinhaltet

- NetFPGA-SUME-Board
- ein mikro-USB-Kabel
- 4 einzigartige MAC Adressaufkleber (eins pro 10G SFP+ Ethernet-Port)



Das NetFPGA-Unterstützungspaket beinhaltet:

- Wiederverwendbare Verilog-Module (IP Cores)
- Überprüfungsinfrastruktur
  - Designsimulation (von der AXI-Schnittstelle)
  - Durchführung von Tests gegen Hardware
  - Testdatengenerierungsbibliotheken
  - Bauen mit xSim und Scapy
  - Benutzung von Python-Skripten für Stimulikonstruktion und Überprüfung
- Infrastruktur bauen

- Bitstream-Generierung durch die Verwendung von Xilinx-Instrumenten
- Registrierungssystem, das Adresse für alle Register und Speicher in einem Projekt generiert. Es benutzt Python- und tcl-Skripte, um HDL-Code und Header-Datei zu generieren.
- Hilfsprogramme
  - Registrierung I/O
- Softwarebibliotheken

## Hardware-Test

NetFPGA-Gruppe entwickelte einige Hardware-Teste, so dass der Benutzer die Hardware sofort überprüfen kann.

## Referenzdesigns

Die NetFPGA-Gemeinschaft bietet eine Reihe von Open-Source-Projekten oder Referenzdesigns. Alles kann direkt vom git-Repository heruntergeladen werden. Um Zugang zu den Referenzdesigns haben zu können, müssen Sie sich unter [http://netfpga.org/site/#/SUME\\_reg\\_form/](http://netfpga.org/site/#/SUME_reg_form/) erfolgreich anmelden. Alle Projekte setzen grundlegende Funktionalitäten eines Schalters, [Routers](#) und [Netzwerk-Interface-Steuereinheit \(NIC\)](#) um und können auf dem Board hochgeladen werden.

Typischerweise besteht ein Projekt oder Referenzdesign aus:

- HDL-Quellen (Verilog/VHDL)
- Simulationstesten
- Hardware-Testen
- Optionaler Software

## Erste Referenz Schaltprojekt

Einige nützliche Referenzdesigns beinhalten Referenz-NIC, Referenzrouter, Referenzschalter und Schalter-Lite. Zum Beispiel, NetSUME-FPGA wird als ein Lernschalter agieren, wenn das Referenzschaltprojekt auf dem Board geladen wird. Die eingehenden Pakete werden zu den entsprechenden Ausgabeporten basierend auf MAC-Adresse übertragen. Bild 1 zeigt die interne Struktur.

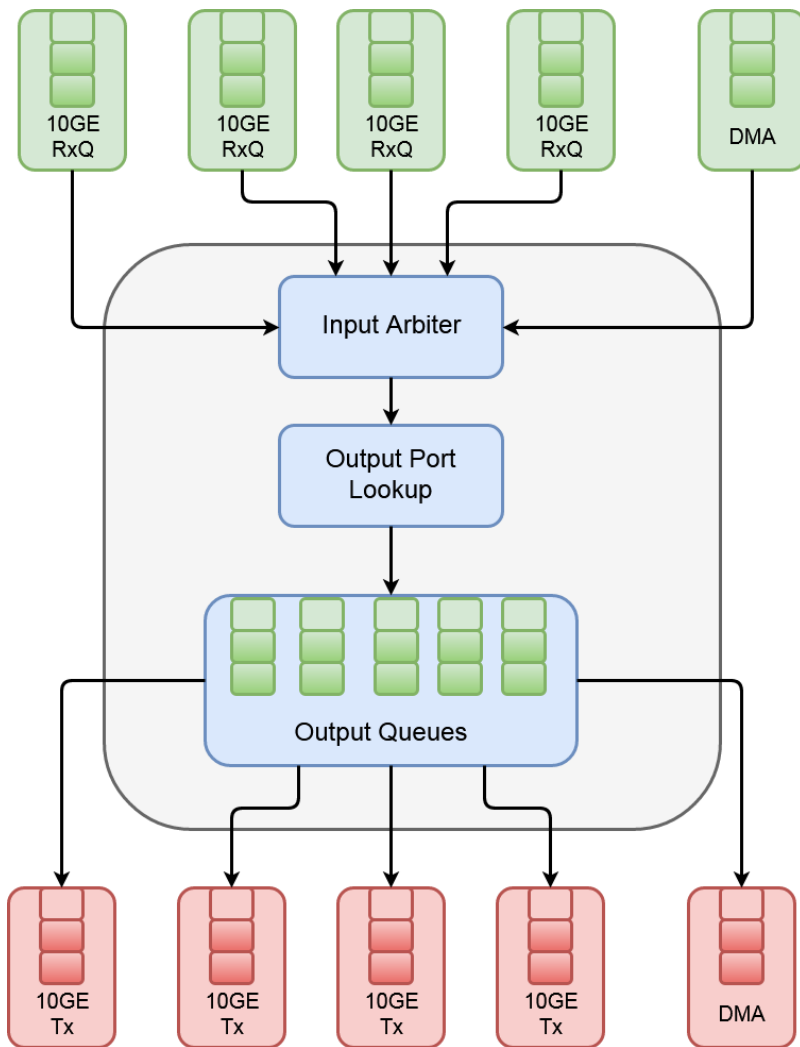


Bild 1

Der Schaltprozess kann in fünf Phasen unterteilt werden:

1. Eingangsport (Rx Queues)
2. Eingangsarbitration
3. Entscheidung und Paketänderung weiterleiten
4. Ausgabewarteschlange
5. Ausgabeport

Nachdem die Benutzer das Grundkonzept und das Grundprinzip verstehen, können sie den Schalter ändern, um die erhaltenen Daten durch die One-Time-Pad Methode: machen Sie ein XOR zwischen der Nachricht und ein Schlüssel K, zu verschlüsseln.

Message :	011010
Key :	110010
	<hr/>
Message XOR Key :	101000
Key :	110010
	<hr/>
Message XOR Key XOR Key :	011010

Die Benutzer können ein Verilog-Modul umsetzen, das die Verschlüsselung und Entschlüsselung der Nutzlast durchführt. Es wurde als ein endlicher Automat (drei Zustände) entworfen, um den Header, der bleibt wie er ist, und die Nutzlast zu identifizieren. Dieses Modul wurde als ein IP verpackt und wird in die Schalterleitung eingeführt. Es gibt zwei Möglichkeiten für die Integration des erhaltenen IP in das Hauptprojekt. Das Blockdiagramm (Blockdesign) ist für jemanden, der Vivado-Blockdesign kennt. Das Tcl-Skript ist gut für die Versionsüberprüfung oder damit der Benutzer Vivado GUI versteht.

1. Benutzen Sie Vivado GUI und Blockdiagramm.
2. Benutzen Sie die Tcl-Skripte, die von der NetFPGA-Gemeinschaft entwickelt wurden.

Nach der Verschlüsselung zeigt Bild 2 die geänderte Struktur - Crypto-Switch.

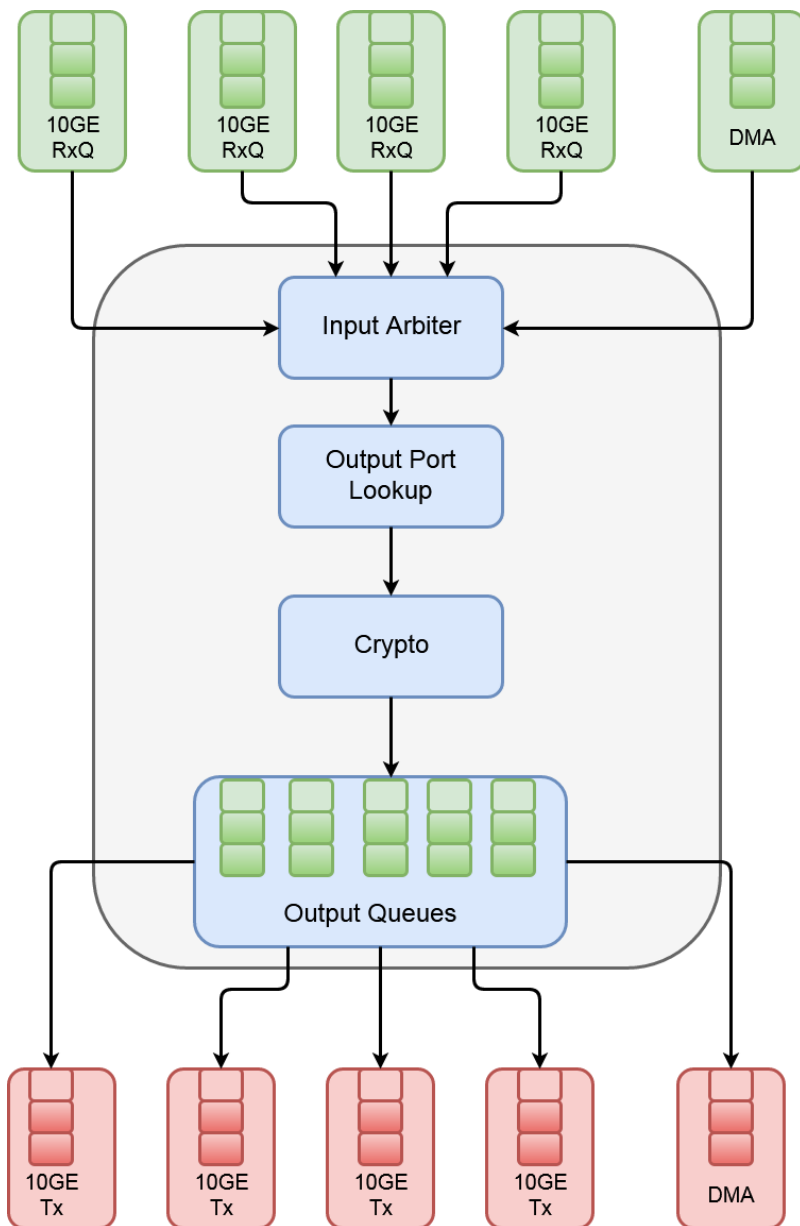


Bild 2

Jetzt können die Benutzer die Simulation durchführen und das Design auf Fehler untersuchen. Die Simulation ermöglicht den Benutzern, das Design zu testen, ohne langwierige Syntheseprozesse zu verlangen. Weil NetSUME-FPGA-Board eine komplexe FPGA-Architektur hat, dauert der Bitstromgenerierungsprozess länger als 45 Minuten. Die Simulation wird im Vivado-Simulator durchgeführt. Um Ethernet-Pakete zu generieren, benutzen wir Scapy. Die Details unter <https://github.com/NetFPGA/NetFPGA-SUME-public/wiki/NetFPGA-SUME-Simulations>.

Sobald die Benutzer das Design durch Simulation validieren, können sie den Bitstrom generieren und auf das Board hochladen. Dieses Vorgehen ist für digitale Designer geeignet, die Verilog oder VHDL kennen und ein Hochleistungssystem als Ziel haben. Ein Software-Ingenieur oder jemand, der die eingehenden Daten analysieren will und aufgrund des Paketheaders oder der Nutzlast handelt, sollte er P4 verwenden. P4 ist eine hochrangige Programmiersprache, die benutzt wird, um die Paketbearbeitungslogik zu beschreiben und Weiterleitungsebene von Netzelementen umzusetzen. Die Benutzer können die Logik in P4 beschreiben und Xilinx SDNet benutzen. Der Code kann unter Xilinx SDNet erarbeitet werden und das erarbeitete Design kann in NetSUME-FPGA hochgeladen werden. Es gibt ein einfaches Schaltbeispiel für NetFPGA SUME in Xilinx SDNet (/opt/Xilinx/SDNet/<version\_number>/data/p4include/sume\_switch.p4). Bild 3 zeigt die Schaltarchitektur.

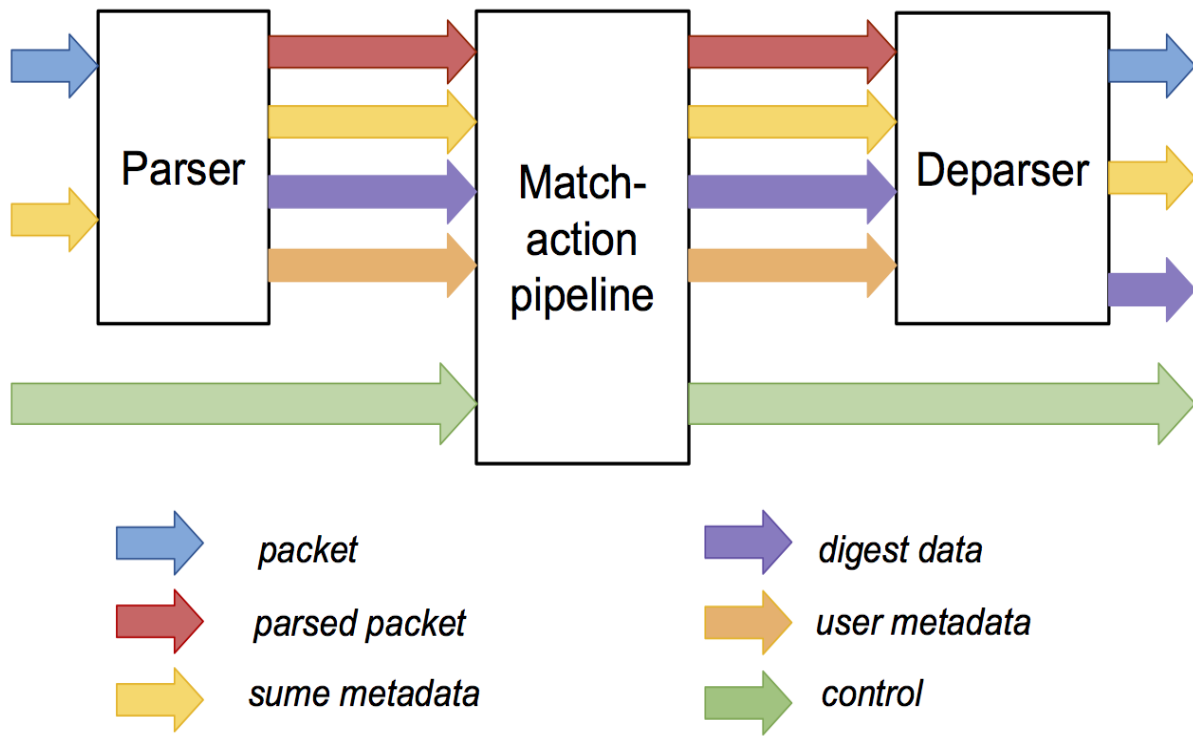


Bild 3

Bild 3

## Referenzen

- NetFPGA Sommerschulkurse: <https://www.cl.cam.ac.uk/research/srg/netos/projects/netfpga/workshop/summer-school-2017/>
- NetFPGA Wiki: <https://github.com/NetFPGA/NetFPGA-SUME-public/wiki>
- NetFPGA SUME repo: <https://github.com/NetFPGA/NetFPGA-SUME-live>