1124 Carrier Kasli 2.0

Features

- 4 SFP 6Gb/s slots for Ethernet and DRTIO
- 12 EEM ports for daughtercards
- 4 MMCX clock outputs
- Xilinx Artix-7 FPGA core
- DDR3 SDRAM

Applications

- Run ARTIQ kernels
- Communicate with the host
- Control other Sinara EEM cards

General Description

The 1124 Kasli 2.0 Carrier card is an 8hp EEM module, designed to run ARTIQ kernels sent from a host machine over the network. It supports up to 12 EEM connections to other EEM cards in the ARTIQ-Sinara family and up to three SFP connections to satellite carriers.

Real-time control of EEM daughtercards is implemented using the ARTIQ RTIO system. 1ns temporal resolution can be achieved for TTL events.

4 SFP Gb/s slots are provided for Ethernet or DRTIO communications. Communication with a host machine is supported over Ethernet, while the Distributed Real-Time Input/Output (DRTIO) system allows for the use of additional core devices (e.g. Kasli 2.0, Kasli-SoC) as satellite cards, capable of running subkernels or distributing commands from the DRTIO master.

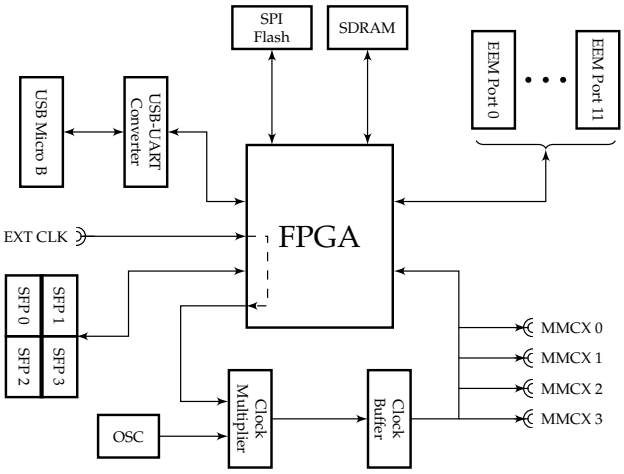

Figure 1: Simplified Block Diagram

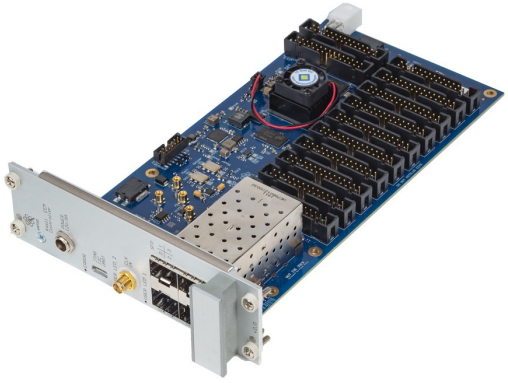

Figure 2: Kasli 2.0 card

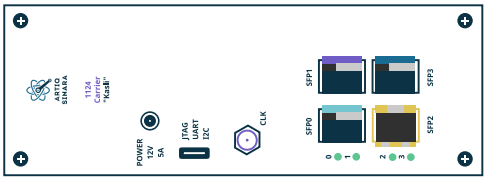

Figure 3: Kasli 2.0 front panel

Source

Kasli 2.0, like all the Sinara hardware family, is open-source hardware, and design files (schematics, PCB layouts, BOMs) can be found in detail at the repository https://github.com/sinara-hw/Kasli.

Electrical Specifications

External clock parameters are derived based on the internal termination specified in UG471[1] and the voltage range specified in DS181[2]. These figures account for the insertion loss of the RF transformer (TC2-1TX+[3]).

Table 1: Recommended Operating Conditions

Parameter	Min.	Typ.	Max.	Unit	Conditions
Clock input					
Input frequency		125		MHz	Si5324 synthesizer bypassed
		10/100/125		MHz	RTIO clock synthesized from input
Power	-9	1.5	5.5	dBm	
Power supply rating		12V, 5A			

Power is to be supplied through the barrel connector in the front panel, size 5.5 mm OD, 2.5 mm ID, and is passed on to daughtercards through the EEM connections. Locking barrel connectors are supported.

FPGA

Kasli 2.0 features an XC7A100T-2FGG484I Xilink Artix-7 FPGA to facilitate reconfigurable high-speed real-time control of inputs and outputs. Most commonly, this FPGA is flashed with binaries compiled from the ARTIQ (Advanced Real-Time Infrastructure for Quantum physics) control system, which equips the carrier board with specialized gateware for handling other Sinara EEMs and an on-FPGA CPU for running ARTIQ experiment kernels.

ARTIQ is open-source and can be found in the repository https://github.com/m-labs/artiq. Orders of Sinara hardware are normally accompanied with precompiled binaries. Long-term support for ARTIQ systems can also be purchased.

A micro-USB located at the back of the Kasli 2.0 board is equipped for JTAG, I2C, and UART serial output.

Note on distributed RTIO (DRTIO)

DRTIO is the time and data transfer system that allows ARTIQ RTIO channels to be distributed among several core device carrier boards, synchronized and controlled by a central core device. The system itself is more fully described in the ARTIQ documentation[4]. With ARTIQ firmware/gateware, supported core devices, including Kasli 2.0, can take one of three roles:

1. **Master**
   The DRTIO master is unique in a DRTIO system. It requires a direct network connection to the host machine. It may make downstream connections to satellites. It controls its own local RTIO channels and the downstream DRTIO satellite(s).

[1] https://docs.xilinx.com/v/u/en-US/ug471_7Series_SelectIO
[2] https://docs.xilinx.com/v/u/en-US/ds181_Artix_7_Data_Sheet
[3] https://www.minicircuits.com/pdfs/TC2-1TX+.pdf
[4] https://m-labs.hk/artiq/manual/drtio.html

2. **Satellite**
   Any other core devices in a DRTIO system are DRTIO satellites. They require an upstream connection to one other core device, master or satellite, through which communications will ultimately be chained to the master. They may make further downstream connections to other satellites. They may control RTIO channels through subkernels or simply pass on communications from the master.

When run in a non-distributed ARTIQ configuration, with a single central core device but no satellites, that core device is instead known as **standalone.**

Communication interfaces

Communication between devices is implemented using 10000Base-X small form-factor pluggable (SFP) interfaces. Four are available on the Kasli 2.0. Appropriate SFP transceivers must be plugged inside the corresponding SFP cages. Each SFP connector possesses an indicator LED.

Upstream connection

A Kasli 2.0 board must acquire an upstream connection through the `SFP0` slot.

- **Standalone/Master**
  An Ethernet-capable SFP transceiver should be inserted into the `SFP0` slot. Typically, a 10000Base-X RJ45 SFP module is used, with an network-connected Ethernet cable attached to the module.
- **Satellite**
  The `SFP0` port should be connected to one of the free SFP slots on an upstream core device, using a cable connection with SFP transceivers.

Downstream Connection

Kasli 2.0 supports up to 3 DRTIO satellite connections per device. Any of the 3 downstream SFP ports (i.e. `SFP1`, `SFP2`, `SFP3`) may be used.

Clock Routing

Standalone/Master

The RTIO clock is typically synthesized by the Si5324 clock multiplier and distributed by the ADCLK948 clock fanout buffer to both the FPGA and the MMCX connectors. Alternatively, an external reference can be supplied through the front panel SMA connector. It is then buffered in the PFGA and sent to the Si5324 for clock synthesis. Kasli 2.0 supports a set of RTIO clock options:

RTIO frequency	Configuration	Clock generation
100 MHz	`int_100`	internal crystal oscillator using PLL, 100 MHz output
125 MHz	`int_125`	internal crystal oscillator using PLL, 125 MHz output (default)
	`ext0_synth0_10to125`	external 10 MHz reference using PLL, 125 MHz output
	`ext0_synth0_100to125`	external 100 MHz reference using PLL, 125 MHz output
	`ext0_synth0_125to125`	external 125 MHz reference using PLL, 125 MHz output
150 MHz	`int_150`	internal crystal oscillator using PLL, 150 MHz output

The clock synthesizer may also be bypassed, using the `ext0_bypass` option, which will accept a RTIO clock directly supplied to the SMA connector. The resulting signal is then routed to both the RTIO system and downstream satellites.

Clocking options in a running system should be configured by setting the value of the `rtio_clock` key to the desired configuration through `artiq_coremgmt`. For example, a RTIO frequency of 125MHz will be synthesized from an external 10 MHz signal after issuing the following command:

```
artiq_coremgmt config write -s rtio_clock ext0_synth0_10to125
```

and rebooting.

Satellite

The RTIO clock is recovered from the SFP transceiver connected to the upstream device. The resulting signal is then cleaned by the Si5324 and routed to both the RTIO system and downstream satellites.

WRPLL

Kasli 2.0 can be configured to use WRPLL, a clock recovery method making use of White Rabbit's DDMTD (Digital Dual Mixer Time Difference) and the card's Si549 oscillator, both to lock the main RTIO clock and to lock satellite clocks to master.

User LEDs

Kasli 2.0 supplies three user LEDs for debugging purposes. Two are located on the front panel. The third is located on the PCB itself, beside the SFP cage. An additional ERR LED on the front panel is used by ARTIQ firmware to indicate a runtime panic.

Example ARTIQ code

The sections below demonstrate simple usage scenarios of extensions on the ARTIQ control system. These extensions make use of the resources on the Kasli 2.0 1124 carrier board. They do not exhaustively demonstrate all the features of the ARTIQ system.

The full documentation for ARTIQ software and gateware, including the guide for its use, is available at https://m-labs.hk/artiq/manual/. Please consult the manual for details and reference material on the functions and structures used here.

Direct Memory Access (DMA)

Instead of directly emitting RTIO events, sequences of RTIO events can be recorded in advance and stored in the local SDRAM. The event sequence can then be replayed at a specified timestamp. This is of special advantage in cases where RTIO events are too closely placed to be generated as they are executed, as events can be replayed at a higher speed than the on-FPGA CPU alone is capable of.

The following example records an LED event sequence and replays it twice consecutively using `CoreDMA`. When run, the LED should blink twice.

```
@kernel
def record(self):
    with self.core_dma.record("led_blink"):
        delay(100*ms)
        self.led0.on()
        delay(100*ms)
        self.led0.off()

@kernel
def playback(self, n):
    handle = self.core_dma.get_handle("led_blink")
    self.core.break_realtime()
    for _ in range(n):
        self.core_dma.playback_handle(handle)

@kernel
def run(self):
    self.core.reset()
    self.record()
    self.playback(2)
```

Stored waveforms can be referenced and replayed in different kernels, but cannot be retrieved and must be regenerated if the core device is rebooted.

Dataset manipulation with core device cache

Experiments may require values computed or found in previously executed kernels. To avoid invoking an RPC or sacrificing the pre-computation in `prepare()` stage, data can be stored in the core device cache.

The following code snippets describe two experiments, in which the data from the first experiment is cached. The data is then retrieved and printed in hexadecimal form in the second experiment.

```
@kernel
def put(self, key, value):
    self.core_cache.put(key, value)

# First experiment
@kernel
def run(self):
    self.put("data", [0xCAFE, 0xDEAD, 0xBEEF])
```

```
@kernel
def get(self, key):
    return self.core_cache.get(key)

@rpc(flags={"async"})
def p(self, p):
    print([hex(_) for _ in p])

# Second experiment
@kernel
def run(self):
    self.p(self.get("data"))
```

Similar to DMA, cached data is no longer retrievable once the core device has been rebooted.

Ordering information

To order, please visit https://m-labs.hk and select 1124 Carrier Kasli 2.0 in the ARTIQ/Sinara crate configuration tool. Cards may also be ordered separately by writing to mailto:sales@m-labs.hk.