ARTIQ Sinara

5716 DAC Shuttler

Features

- 16-channel DAC
- 14-bit resolution, < 1 LSB DNL
- 125 MSPS sample rate
- Output voltage ±10 V
- EEM FMC carrier with Artix-7 FPGA core
- Remote analog front end card

Applications

- Driving DC electrodes in ion traps
- Ion chain splitting, ion shuttling

General Description

The 5716 DAC Shuttler is an 8hp EEM module, shipped with associated remote analog front-end (AFE), part of the ARTIQ/Sinara family. It consists of the Shuttler FMC paired with an 8hp Sinara EEM FMC Carrier, which is capable of running as an ARTIQ satellite core through DRTIO-over-EEM. It adds digital-analog conversion capabilities to carrier cards such as 1124 Kasli and 1125 Kasli-SoC.

ARTIQ gateware implements NIST PDQ-style waveform synthesizer which supports the use of sigma-delta modulation to increase effective resolution to 16 bits.

Digital communication between FMC and remote AFE is provided through mini-SAS HD cables. The AFE supports ±10 V output and 50 MHz 3dB bandwidth, using onboard 24-bit ADC for calibration.

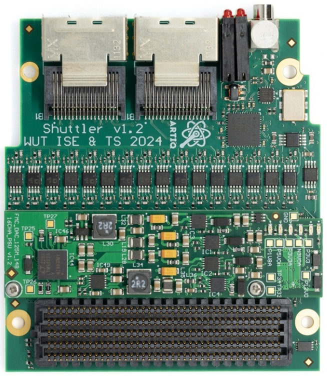


Figure 1: Shuttler FMC

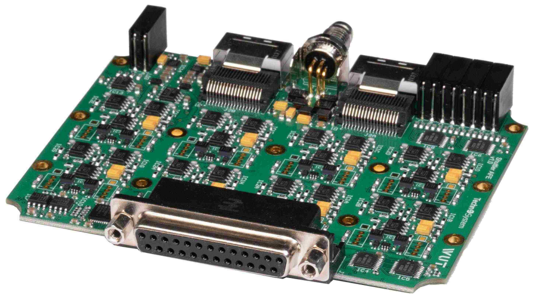


Figure 2: Shuttler AFE

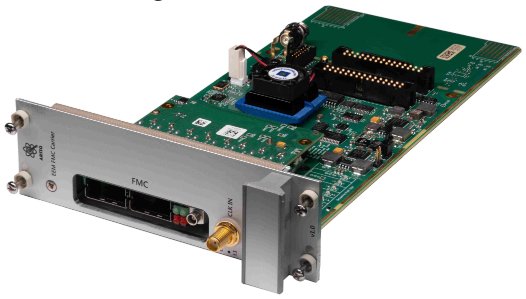


Figure 3: Sinara EEM FMC carrier

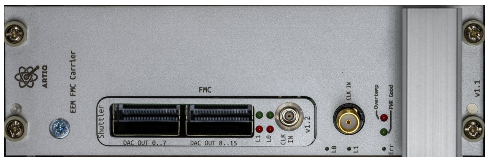


Figure 4: Shuttler front panel

Source

5716 DAC Shuttler, 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/FMC_Shuttler. Files for the AFE card are stored at https://github.com/sinara-hw/AFE_DAC_External. Files for the Sinara EEM FMC Carrier can be found at https://github.com/sinara-hw/EEM_FMC_Carrier.

Electrical Specifications

These specifications are based on the datasheet of the DAC IC (AD9117[1]), board measurements[2], and various information from the Sinara wiki[3].

Table 1: Output Specifications

Parameter	Min.	Typ.	Max.	Unit	Conditions
Sampling rate[3]		125		MSPS	
Output voltage[3]	-10		+10	V	
Resolution[3]		14		bits	Raw
		16		bits	With sigma-delta modulation
Settling time[1]		11.5		ns	
Analog bandwidth[2]		12		MHz	
3dB bandwidth[3]		50		MHz	

Power to Shuttler is supplied over EEM. Power to the AFE is to be supplied over a 4-pin circular M8 connector placed between the mini-SAS HD ports. The AFE output port is 25-pin DSUB.

Firmware/ARTIQ

ARTIQ is open-source and can be found in the repository https://github.com/m-labs/artiq. Orders of Sinara hardware are normally preflashed with suitable firmware and gateware binaries. Long-term support for ARTIQ systems can also be purchased, including updated binaries through AFWS (the ARTIQ Firmware Service).

The Sinara EEM FMC Carrier features an XC7A200T-3FBG484E Xilinx Artix-7 FPGA, usually configured as an ARTIQ satellite core. Firmware and gateware for the Sinara EEM FMC Carrier is closely related to that used for 1124 Kasli 2.0 satellites. The specific binary generation target can be found in the module `artiq.gateware.targets.efc` of the ARTIQ repository.

[1] https://www.analog.com/media/en/technical-documentation/data-sheets/AD9114_9115_9116_9117.pdf
[2] https://github.com/sinara-hw/FMC_Shuttler/issues/36
[3] https://github.com/sinara-hw/FMC_Shuttler/wiki

ARTIQ System Description Entry

ARTIQ/Sinara firmware/gateware is generated according to a JSON system description file, allowing gateware to be specific to and optimized for a certain system configuration.

5716 Shuttler should be entered in the `peripherals` list of the corresponding core device in the following format:

```
{
    "type": "shuttler",
    "ports": 0
}
```

Replace 0 with the EEM port number used on the core device. Any port can be used. On the other side, the Sinara EEM FMC Carrier possesses two EEM ports, but only one is necessary for Shuttler. This should always be `EEM0`.

Since Shuttler acts as a DRTIO satellite, the DRTIO type of the core device should be specified as master, not standalone, even if no other satellite cores are used. DRTIO-over-EEM for Shuttler is automatically assigned a destination number, #4 on Kasli 2.0, #5 on Kasli-SoC[4]. Destination numbers count up correspondingly for additional Shuttlers. See the ARTIQ manual[5] for instructions on configuring a routing table, for cases where you need one (for example, a Shuttler on a DRTIO satellite).

Clocking

Clock input should be provided to Shuttler through the EEM FMC Carrier. The EEM FMC Carrier *must* share a clock source with the associated core device. Clocks must be aligned to utilize DRTIO-over-EEM. Clock input can be provided to EEM FMC Carrier via SMA connector on front panel or MMCX connector at back of board (top right, above `EEM0`). The Shuttler FMC features a front panel MCX connector labeled for clock input; this is currently unused by ARTIQ firmware/gateware.

FMC Carrier clock source must be configured by setting the DIP switches on back of the board, under the following schema:

Clock Source	CLK_SEL0	CLK_SEL1
Front panel SMA	0	0
Internal oscillator	1	0
Back MMCX	0	1
PE CLK	1	1

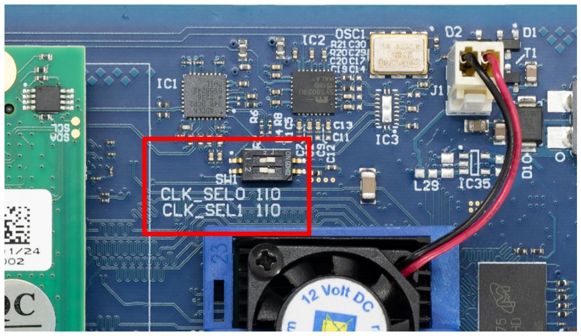


Figure 5: Position of DIP switches

Users should note that PE CLK and internal oscillator are not valid source choices for Shuttler.

At first power-up, FMC Carrier and connected core device will determine the clock skew over EEM transceiver and store the result in configuration memory. It can be accessed in ARTIQ under the key `eem_drtio_delay0` (where `0` is a counter that will be incremented for further DRTIO-over-EEM connections.)

If EEM cable or clocking cables are changed, or if either device is reflashed for any reason, this value must be manually erased in order to force a reevaluation of the clock skew. Either `artiq_coremgmt config remove` (for original ARTIQ) or direct access to the SD card (on Zynq) should be used.

[4] i.e., in both cases, first available destination number after those associated with the core device's downstream SFP slots.
[5] https://m-labs.hk/artiq/manual/using_drtio_subkernels.html

LEDs

The EEM FMC Carrier provides two user LEDs, `L0` and `L1`, located on the front panel, which are accessible in ARTIQ gateware and can be used for testing.

The Shuttler AFE provides twenty LEDs in two banks. The four-LED bank to the right of the mini-SAS connectors indicate power status. The sixteen-LED bank to the left of the mini-SAS connectors indicate output relay status. DAC output is only valid when corresponding relay LEDs are on.

Example ARTIQ Code

The sections below demonstrate simple usage scenarios of extensions on the ARTIQ control system. These extensions make use of the resources of the 5716 DAC Shuttler. They do not exhaustively demonstrate all the features of the ARTIQ system.

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

Shuttler is capable of generating a waveform in the following equation:

$$w(t) = a(t) + b(t) * cos(c(t))$$

where $a(t)$ and $b(t)$ are cubic splines and $c(t)$ is a quadratic spline[6].

The following code initializes relay and ADC and resets all channels.

```
@kernel
def relay_init(self):
    self.shuttler0_relay.init()
    self.shuttler0_relay.enable(0x0000)

@kernel
def adc_init(self):
    delay_mu(int64(self.core.ref_multiplier))
    self.shuttler0_adc.power_up()

    delay_mu(int64(self.core.ref_multiplier))
    assert self.shuttler0_adc.read_id() >> 4 == 0x038d

    delay_mu(int64(self.core.ref_multiplier))
    # The actual output voltage is limited by the hardware,
    # the calculated calibration gain and offset.
    # For example, if the system has a calibration gain of
    # 1.06, then the max output voltage = 10 / 1.06 = 9.43V.
    # Setting a value larger than 9.43V will result in overflow.
    self.shuttler0_adc.calibrate(
        self.shuttler0_dcbias, self.shuttler0_trigger, self.shuttler0_config)
```

[6]See also the PDQ documentation hosted at the following link: https://pdq.readthedocs.io/

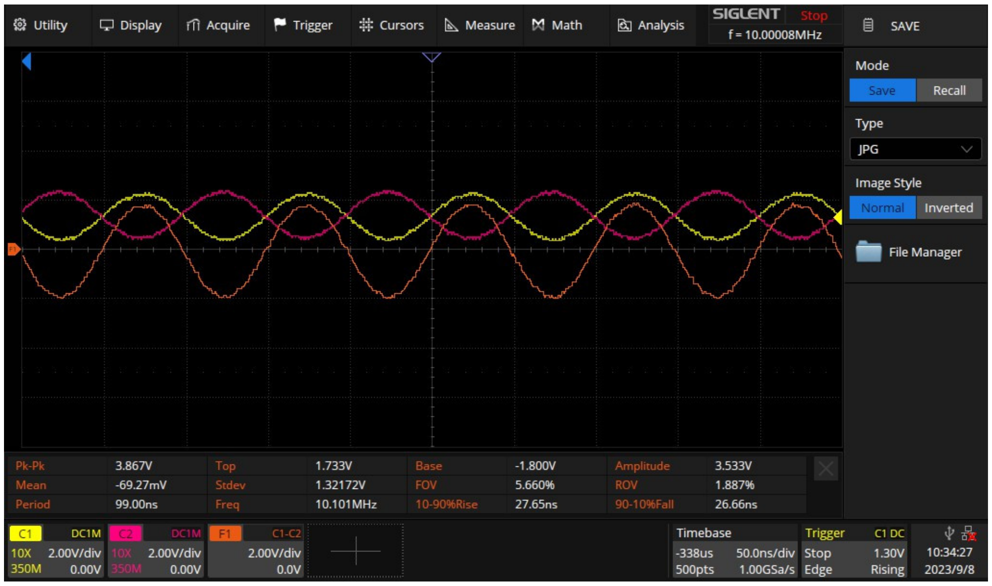
```
@kernel
def shuttler_channel_reset(self, ch):
    self.shuttler0_dcbias[ch].set_waveform(
        a0=0, a1=0, a2=0, a3=0,
    )
    self.shuttler0_dds[ch].set_waveform(
        b0=0, b1=0, b2=0, b3=0,
        c0=0, c1=0, c2=0,
    )
    self.shuttler0_trigger.trigger(1 << ch)

@kernel
def run(self):
    self.core.reset()
    self.core.break_realtime()

    self.relay_init()
    self.adc_init()

    for i in range(16):
        self.shuttler_channel_reset(i)
        # To avoid RTIO Underflow
        delay(50*us)
```

Generating a basic waveform

The following code generates a basic sine wave of approx 10 MHz on the `DAC0 I` channel. The value of `0x147AE148` used for c_1 sets the frequency as $c_1/2^{32} * 125$ MHz.

```
@kernel
def sine(self):
    for i in range(2):
        self.shuttler0_dcbias[i].set_waveform(
            a0=0,
            a1=0,
            a2=0,
            a3=0,
        )
        self.shuttler0_dds[i].set_waveform(
            b0=0x0FFF,
            b1=0,
            b2=0,
            b3=0,
            c0=0,
            c1=0x147AE148, # Frequency = 10MHz
            c2=0,
        )
    self.shuttler0_trigger.trigger(0xFFFF)
```

For more example waveforms see also the folder `kasli_shuttler` in the ARTIQ `examples` directory.



f = 10.00008MHz

Mode: Save
Type: JPG
Image Style: Normal

Pk-Pk	3.867V	Top	1.733V	Base	-1.800V	Amplitude	3.533V
Mean	-69.27mV	Stdev	1.32172V	FOV	5.660%	ROV	1.887%
Period	99.00ns	Freq	10.101MHz	10-90%Rise	27.65ns	90-10%Fall	26.66ns

```
C1    DC1M        C2    DC1M        F1     C1-C2
10X   2.00V/div   10X   2.00V/div          2.00V/div
350M  0.00V       350M  0.00V              0.0V

Timebase                Trigger   C1 DC
-338us    50.0ns/div    Stop      1.30V     10:34:27
500pts    1.00GSa/s     Edge      Rising    2023/9/8
```

Figure 6: Produced waveform, measured at `AFE0` output resistor R36A, R39A.

Ordering Information

To order, please visit `https://m-labs.hk` and choose 5716 DAC Shuttler in the ARTIQ/Sinara hardware selection tool. Cards can be ordered as part of a fully-featured ARTIQ/Sinara crate or standalone through the 'Spare cards' option. Otherwise, orders can also be made by writing directly to `mailto:sales@m-labs.hk`.