RAPTOR

32-bit Arm Cortex M0 MCU

Features

- Core: ARM 32-bit Cortex-M0 processor, frequency up to 50 MHz
- Memory: 16 kB SRAM
- QSPI XiP memory controller for external Flash memory

- Reset and power management
  - Single power supply (3.3V)
  - On-board 1.8V LDO regulator
  - 3.3V I/O
  - Power-on (POR)
  - Low power mode thru clock freq scaling

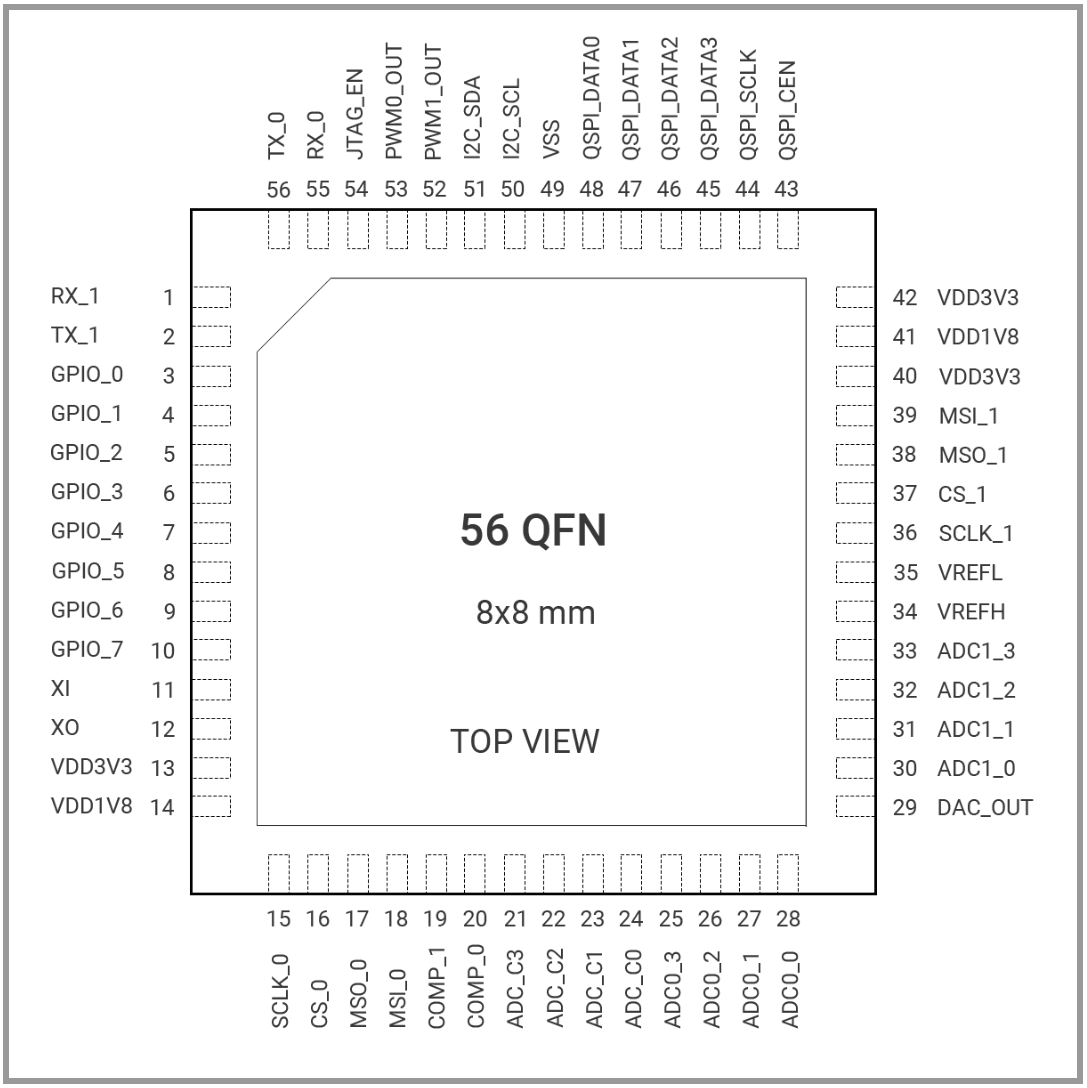
- Clock management
  - Software configurable internal and external clock sources
    - 1 to 4 MHz crystal oscillator
    - 10 MHz RC oscillator
  - Pre-designed clocking subsystem including 8x PLL and clock divider

- Analog I/O
  - Two 10-bit, 1.0 μs ADCs (up to 8 channels each)
    - 4 channels common
  - One 10-bit DAC channel
  - One analog comparator

- Peripherals
  - Two multi-function 32-bit PWM/timers
  - One SPI master controller (AHB bus)
  - One SPI master controller (APB bus)
  - One 8-bit GPIO port
  - One I2C master controller - standard mode
  - Two UART with FIFO and baud rate generator

- JTAG port for testing
- 56-pin QFN package (8x8m)

Pin Diagram



Standard package:	QFN 56
Package size:	8mm x 8mm
Pin pitch:	0.5 mm
Paddle size:	6.55mm x 6.55 mm

Pinout Descriptions

Name	Pin(s)	I/O	Description
TX_0	56	O	UART transmit channel
RX_0	55	I	UART 0 receive channel
TX_1	1	O	UART transmit channel
RX_1	2	I	UART 1 receive channel
GPIO[7:0]	3-10	I/O	General purpose configurable digital I/O with pullup/pulldown, input or output, and enable/disable.
XI	11	I	Crystal oscillator input
XO	12	O	Crystal oscillator output
VDD3V3	13, 40, 42	I	3.3V Power (in)
VDD1V8	14, 41	O	1.8V Power (out)
VSS	49	I	Ground + Paddle
PWM0_OUT	53	O	PWM 0 output
PWM1_OUT	52	O	PWM 1 output
I2C_SDA	51	I/O	I2C data
I2C_SCL	50	I/O	I2C clock
MSI_0	39	I	Serial interface 0 data input
MSO_0	38	O	Serial interface 0 data output
CS_0	37	O	Serial interface 0 chip select
SCLK_0	36	O	Serial interface 0 clock
MSI_1	18	I	Serial interface 1 data input
MSO_1	17	O	Serial interface 1 data output
CS_1	16	O	Serial interface 1 chip select
SCLK_1	15	O	Serial interface 1 clock
QSPI_DATA[3:0]	48-45	I/O	QSPI flash bidirectional data input/output
QSPI_SCLK	44	O	QSPI chip select
QSPI_CEN	43	O	QSPI clock

Name	Pin(s)	I/O	Description
ADC0 [3:0]	28-25	I(A)	ADC 0 input
ADC1 [3:0]	33-30	I	ADC 1 input
ADC_C [3:0]	24-21	I	ADC common inputs (between ADC 0 and ADC 1)
COMP_0	20	I	Comparator positive input
COMP_1	19	I	Comparator negative input
DAC_OUT	29	O	DAC output
VREFH	34	I	ADC and DAC voltage reference (high)
VREFL	35	I	ADC and DAC voltage reference (low)
JTAG_EN	54	I	JTAG enable

Overview

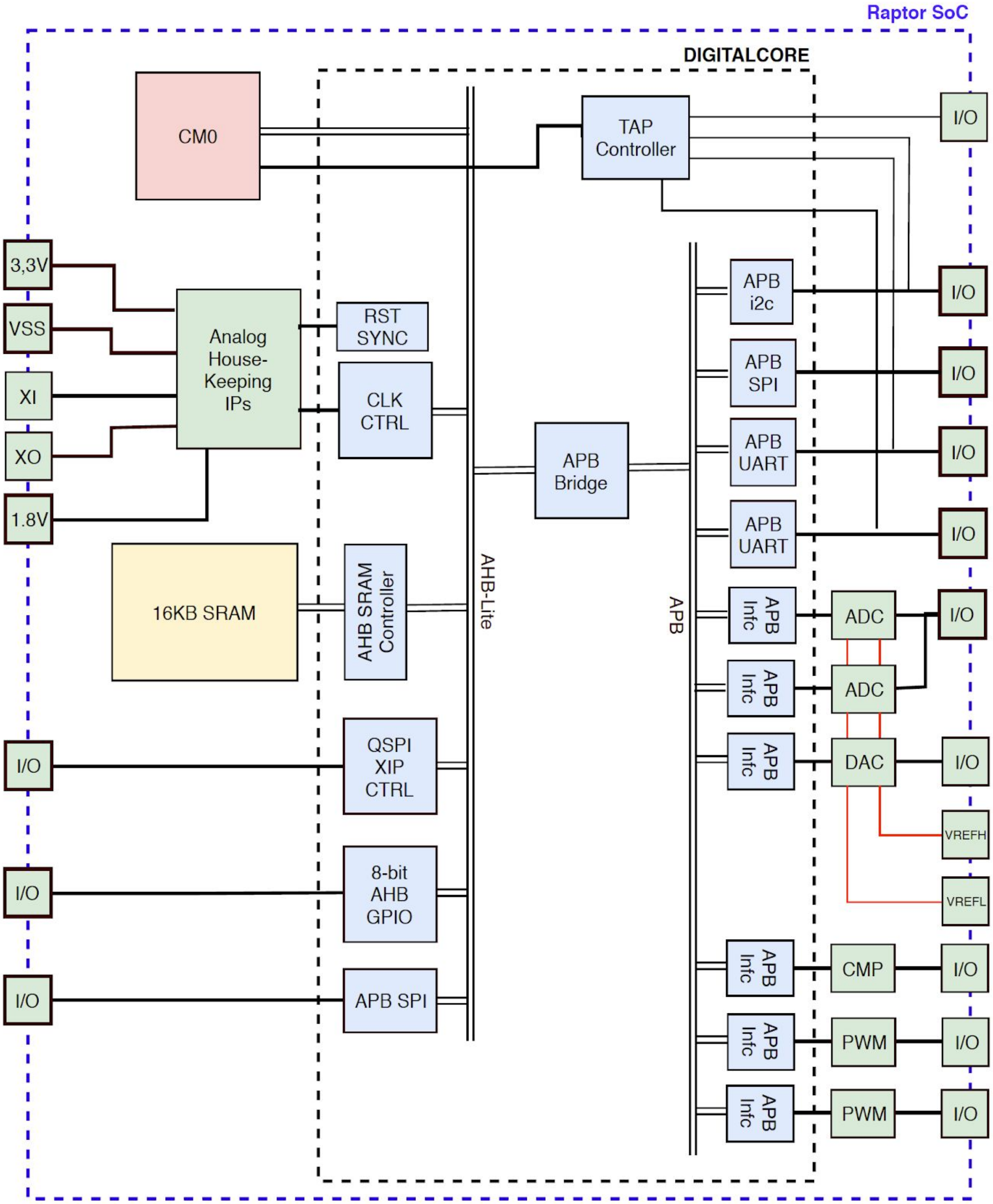
Raptor is a full SoC reference design for IoT applications based on the Arm Cortex M0 or M3 CPU core targeted on the X-FAB XH018 process (180nm). Raptor provides means to interface with analog and digital sensors as well as external RF radio (WIFI, Cellular, Bluetooth, etc.,) modules needed for IoT applications.

The reference design is based on an SoC design template and can be modified, compiled and simulated through the SoC Editor tool with CloudV on the Efabless platform. The reference design comes with basic device drivers for all on-chip peripherals as well as set of applications to test and demonstrate Raptor capabilities.

Instances of the Raptor Design Template can be configured through the Design Request Form. A configuration file is generated upon saving the form and can be used as a starting point for generating a reference design or to request a turn-key delivery of a custom SoC through design partners on the Efabless platform.

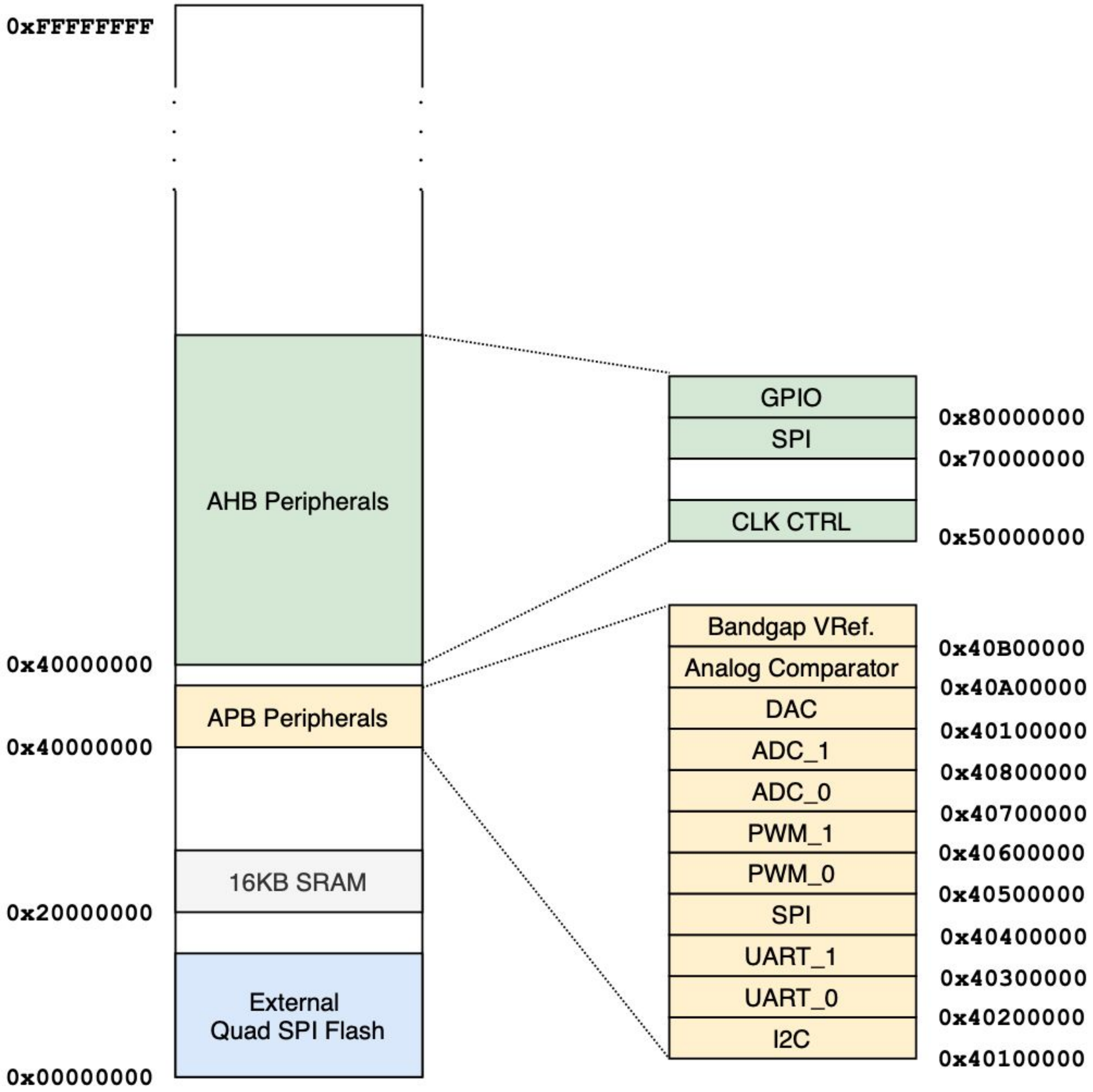
This datasheet provides specifications for the Demo chip configuration tape-out for Raptor. The complete project including layout for this configuration can be cloned from the Raptor entry in the design catalog.

Block Diagram



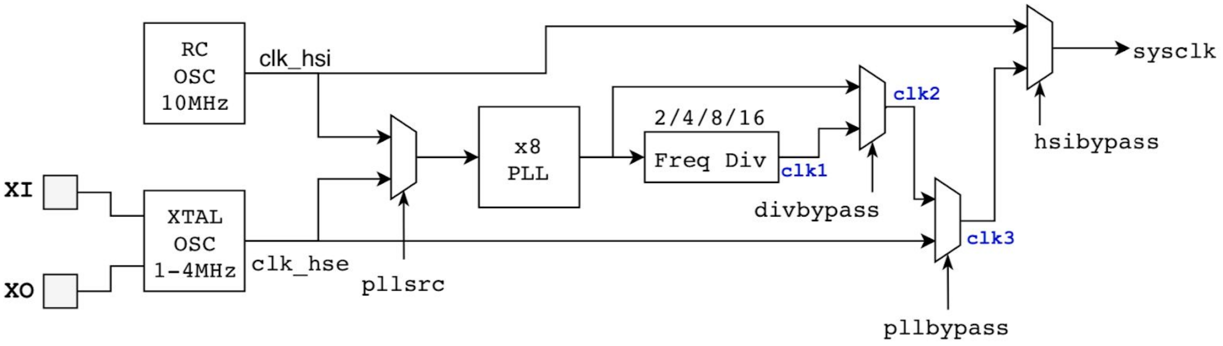
Memory Organization

Raptor has a 16KB SRAM block located at address 0x20000000. Also, it has eXecute In Place QUAD SPI Flash interface which is located at address 0x00000000. The peripherals I/O registers are located at 0x40000000. The following figure shows the memory map.



Clocking

Raptor can be operated using external or internal clock sources. Raptor has 10MHz internal RC oscillator (RCOSC) as well as the necessary circuitry to attach external crystal oscillator (XTOSC) that needs an external crystal or ceramic resonator (1-4 MHz) and a pair of capacitors. Also, Raptor employs a PLL and a clock divider to generate system clock frequencies from 500 MHz (using external crystal of 1MHz) to 80MHz[1] (generated from the internal 10MHz RCOSC with 8x PLL) under software control. The following figure illustrates Raptor clocking sub-system.



Raptor clocking sub-system can be configured during run-time through the following I/O registers (Base address: 0x0x50000000):

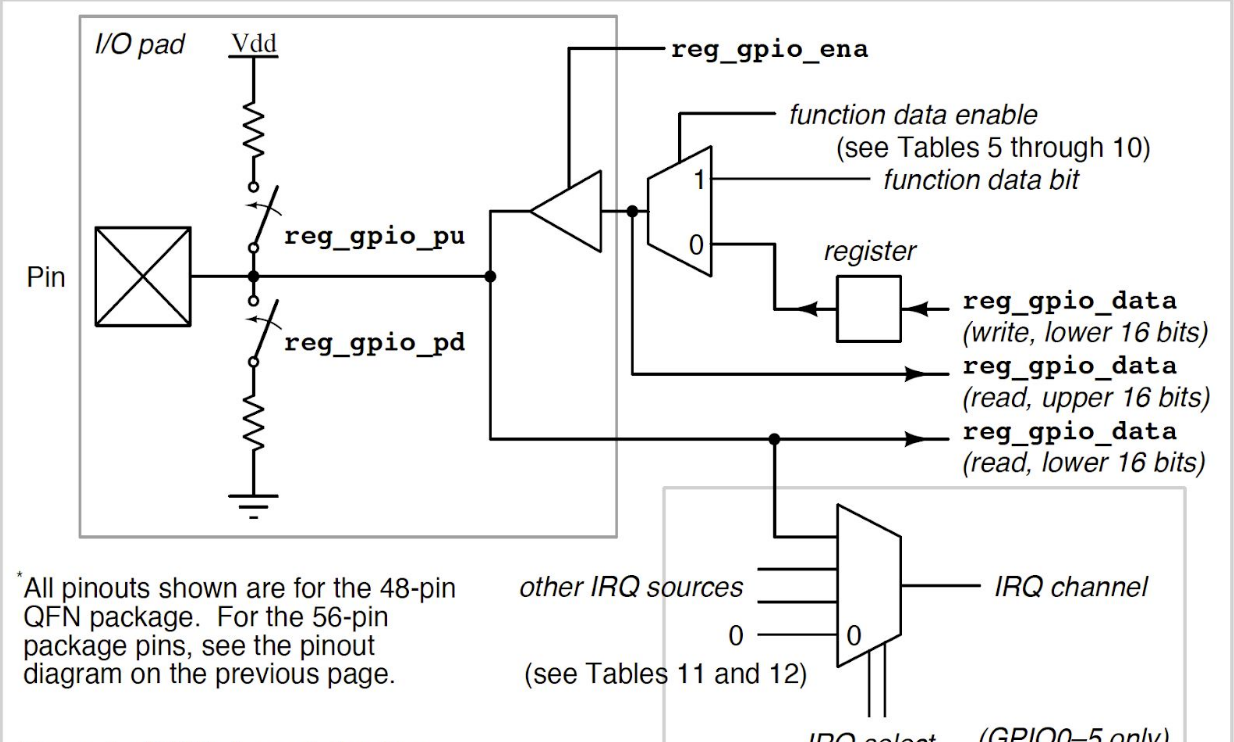
Register	Offset	Function
PLLCR	0	PLL Control Register 0: PLL enable[a] 1: pllsrc (PLL Source) - 0: RCOSC, 1: XTOSC 2-3: PLL divisor (00:2, 01:4, 10:8 or 11:16)
PLLTR	4	PLL Trim Register 0: Enable VCO trimming 1-4: VCO trim value
CLKCR	8	Clock Control Register 0: pllbypass (PLL Bypass) - 1: XTOSC, 0: PLL output 1: divbypass (Divisor Bypass) - 1: bypass the PLL divisor 2: hsibypass (RCOSC Select - inverted) - 0: sysclk is connected to the RCOSC 3: rcoscen (RCOSC Enable - Inverted) - 0: Enabled 4: xtoscen (XTOSC Enable) - 1: Enabled

[a] The PLL needs at least 25usec to stabilize once enabled.

[1] Raptor was designed to run at frequencies up to 50MHz. Any configuration that generates a clock higher than 50MHz must not be used.

GPIO

The GPIO pins are sixteen assignable digital inputs or outputs. The basic function of each GPIO is illustrated below. All writes to reg_gpio_data are registered. All reads from reg_gpio_data are immediate.



*All pinouts shown are for the 48-pin QFN package. For the 56-pin package pins, see the pinout diagram on the previous page.

Figure 1. GPIO channel structure

GPIO memory address map:

C header name	address	description
reg_gpio_data	0x80000000	GPIO input/output (lower 16 bits) GPIO output readback (upper 16 bits)
reg_gpio_enb	0x80000004	GPIO output enable (0 = output, 1 = input)
reg_gpio_pub	0x80000008	GPIO pullup enable (0 = pullup, 1 = none)
reg_gpio_pdb	0x8000000c	GPIO pulldown enable (0 = pulldown, 1 = none)

SPI

Raptor has two SPI master controller modules:

- SPI0 connected to the APB bus (base address: 0x40400000) and the external pins MSI_0, MSO_0, CS_0 and SCLK_0.
- SPI1 connected to the AHB-Lite bus (base address: 0x70000000) and the external pins MSI_1, MSO_1, CS_1 and SCLK_1.

Each of the SPI master controller modules has 4 registers as given by the following table.

Register	Offset	Function
SPIDATA	0x0	SPI Data register
SPICTRL	0x4	SPI control register 0: Start 1: Slave Select (SS0 bit)
SPICFG	0x8	SPI Configuration Register 0: SPI Clock Polarization (CPOL) 1: SPI Clock Phase (CPHA) 2-9: SPI clock prescale
SPISTATUS	0x10	SPI Status Register 0: Done

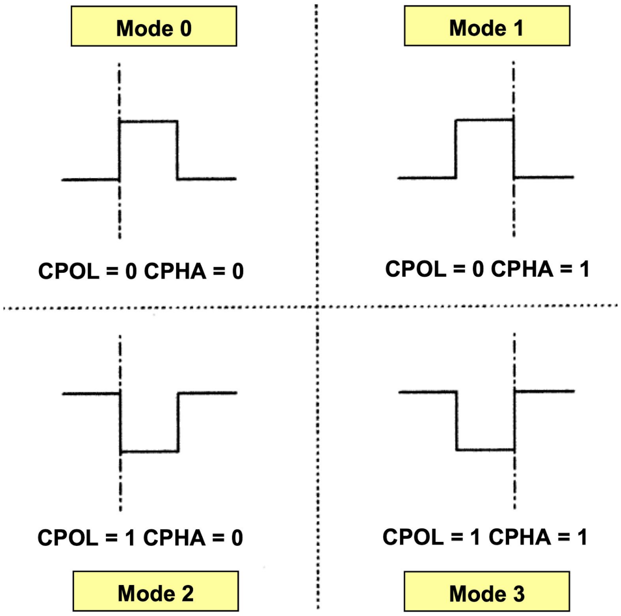


Figure 1. SPI modes

Phase and Polarity

The master controller can select the clock polarity and clock phase. The CPOL bit in the SPICFG register sets the polarity of the clock signal during the idle state. The idle state is defined as the period when CS is high and transitioning to low at the start of the transmission and when CS is low and transitioning to high at the end of the transmission. The CPHA bit selects the clock phase. Depending on the CPHA bit, the rising or falling clock edge is used to sample and/or shift the data. The master must select the clock polarity and clock phase, as per the requirement of the slave. Depending on the CPOL and CPHA bit selection, four SPI modes are available. Figure 1 shows the four SPI modes.

Bit Rate

SPI clock is derived from the system clock. An 8-bit prescaler register can be used to divide the clock frequency to generate the desired bit rate. The minimum value of the clock divider is 4.

$$SPI\ Clock\ Frequency\ =\ \frac{System\ Clock}{SPICFG[9{:}2]}$$

Transmitting a byte

Receiving a byte

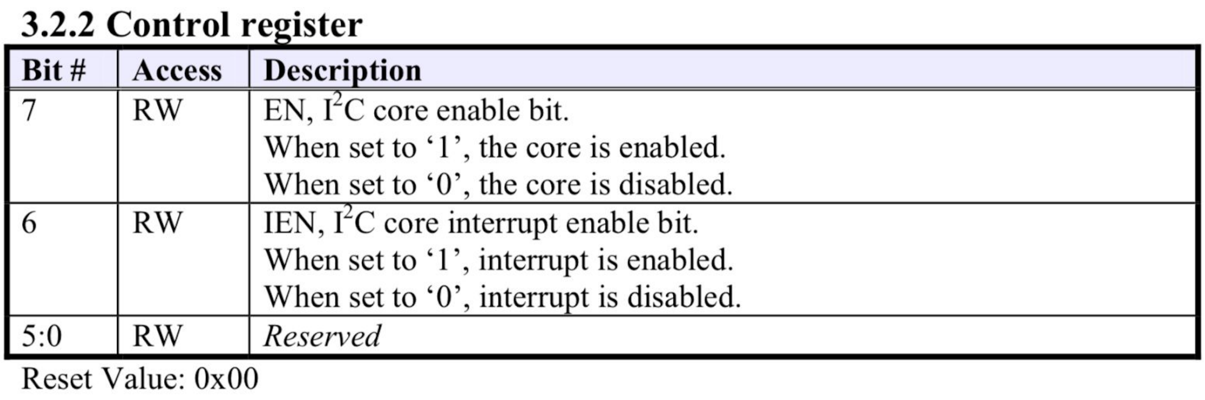
I2C

Raptor has one I2C controller module on the APB bus (base address: 40100000). The I2C module has the following features:

- Software programmable clock frequency
- Clock Stretching and Wait state generation
- Software programmable acknowledge bit
- Start/Stop/Repeated Start/Acknowledge generation
- Start/Stop/Repeated Start detection
- Bus busy detection
- Supports 7 and 10bit addressing mode

Prescale Register

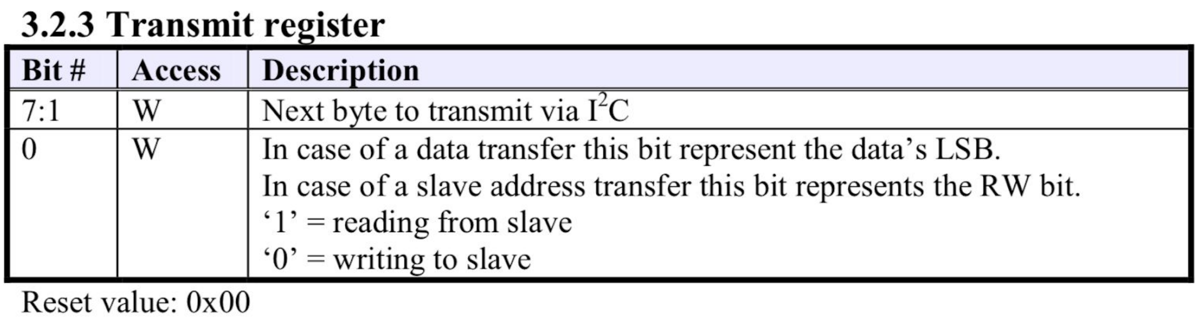
This register is used to prescale the SCL clock line. Due to the structure of the I2C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when the 'EN' bit is cleared.



3.2.2 Control register

Bit #	Access	Description
7	RW	EN, I^{2}C core enable bit. When set to '1', the core is enabled. When set to '0', the core is disabled.
6	RW	IEN, I^{2}C core interrupt enable bit. When set to '1', interrupt is enabled. When set to '0', interrupt is disabled.
5:0	RW	*Reserved*

Reset Value: 0x00



3.2.3 Transmit register

Bit #	Access	Description
7:1	W	Next byte to transmit via I^{2}C
0	W	In case of a data transfer this bit represent the data's LSB. In case of a slave address transfer this bit represents the RW bit. '1' = reading from slave '0' = writing to slave

Reset value: 0x00

3.2.4 Receive register

Bit #	Access	Description
7:0	R	Last byte received via I^2C

Reset value: 0x00

3.2.5 Command register

Bit #	Access	Description
7	W	STA, generate (repeated) start condition
6	W	STO, generate stop condition
5	W	RD, read from slave
4	W	WR, write to slave
3	W	ACK, when a receiver, sent ACK (ACK = ‘0’) or NACK (ACK = ‘1’)
2:1	W	*Reserved*
0	W	IACK, Interrupt acknowledge. When set, clears a pending interrupt.

Reset Value: 0x00

3.2.6 Status register

Bit #	Access	Description
7	R	RxACK, Received acknowledge from slave. This flag represents acknowledge from the addressed slave. ‘1’ = No acknowledge received ‘0’ = Acknowledge received
6	R	Busy, I^2C bus busy ‘1’ after START signal detected ‘0’ after STOP signal detected
5	R	AL, Arbitration lost This bit is set when the core lost arbitration. Arbitration is lost when: • a STOP signal is detected, but non requested • The master drives SDA high, but SDA is low. See *bus-arbitration* section for more information.
4:2	R	*Reserved*
1	R	TIP, Transfer in progress. ‘1’ when transferring data ‘0’ when transfer complete
0	R	IF, Interrupt Flag. This bit is set when an interrupt is pending, which will cause a processor interrupt request if the IEN bit is set. The Interrupt Flag is set when: • one byte transfer has been completed • arbitration is lost

Reset Value: 0x00

UART

Raptor has 2 simple Universal Asynchronous Receiver/Transmitter (UART0 and UART1) that are connected to the APB bus. UART0 is located at address 0x40200000 and UART1 at address 0x40300000. Both UART modules have the following characteristics:

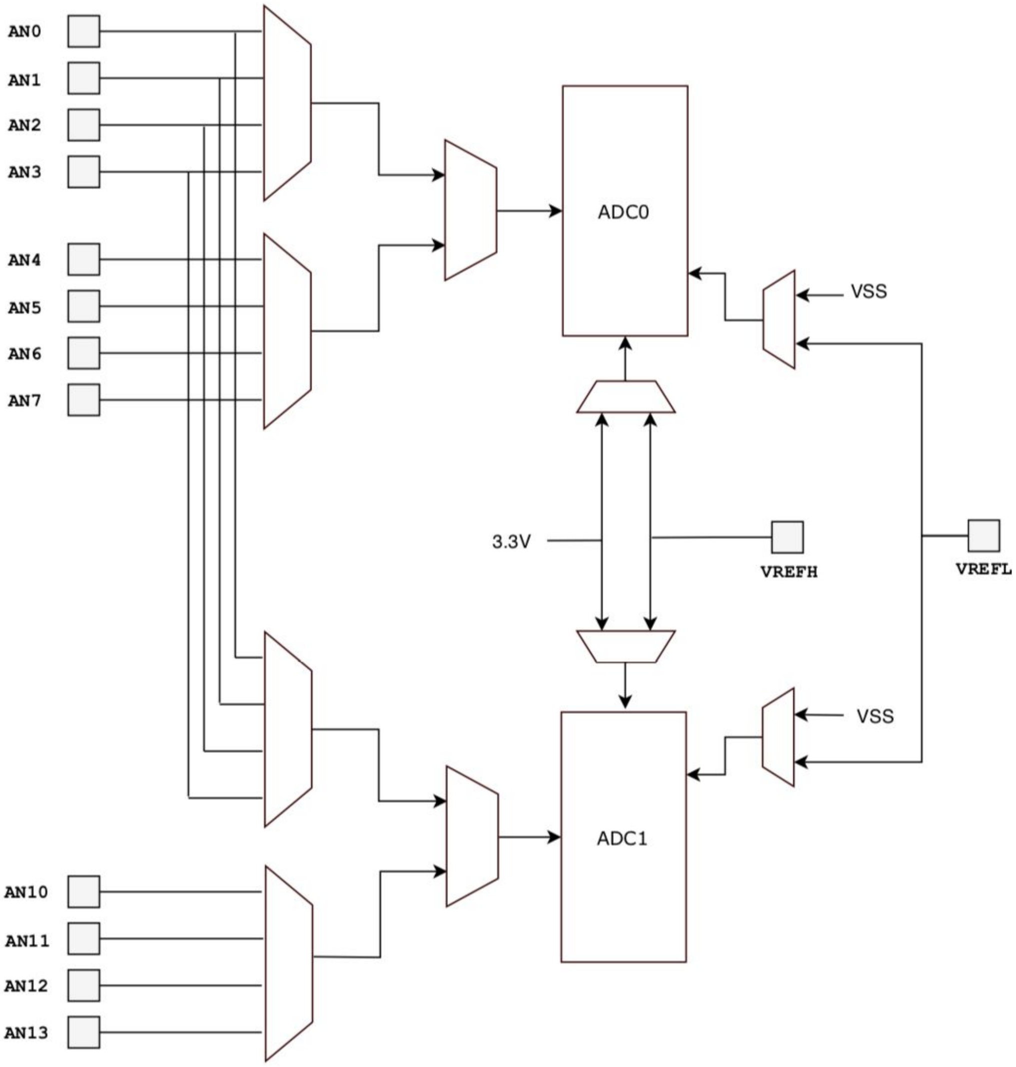
- Baud rate is sysclk/163*16. For 50MHz, the baud rate is 19200.
- Data bits: 8
- Parity: None
- Stop bits: 1

Each UART uses two FIFOs as buffers for both sending and receiving data. Reading from the FIOFO while being empty would block the execution till the UART receives data. Same behavior is observed when trying to write to the transmission FIFO while being full. The UART generates an interrupt, which is driven HIGH whenever there is data to be read on the receiving FIFO. The interrupt is cleared when the data is read. UART0 uses IRQ0 (INT16) and UART1 uses IRQ1 (INT17).

PWM

ADC

Raptor employs two 10-bit Successive Approximation (SAR) Analog to Digital Converters (DAC0 and DAC1) with 12 analog input channels. Each DAC has 4 dedicated channels (AN4-AN7 for DAC0 and AN8-AN11 for DAC1). The remaining 4 channels are shared between the 2 DACs as illustrated by the following figures.



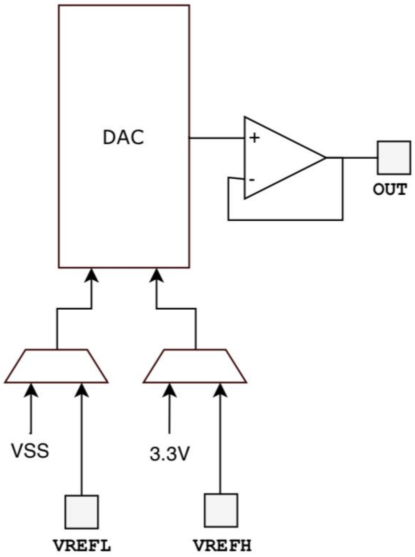
The analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and VSS), or the voltage level on the VREFH and VREFL pins.

ADC0 and ADC1 modules are connected to the APB bus and located at 0x40700000 and 0x40800000 respectively. The following are the I/O register for any of the ADC modules:

Register	Offset	Function
ADCCTRL	0x00	ADC control register 0: ADC Enable 1: Start Conversion 2-4: Channel selection 5: VREF Select - 0: External (VREFH, VREFL) 1: Internal (VDD, VSS)
ADCDATA	0x04	0-9: Data
ADCSTATUS	0x08	0: End of conversion flag
ADCPRESCALER	0x10	Prescaler for ADC clock ADC clock = Sysclk/(2*(ADCPRESCALER + 1))

DAC

Raptor has one 10-bit Digital to Analog Converter (DAC) module with a buffered output (using a voltage follower) as shown in the following Figure.



The analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and VSS), or the voltage level on the VREFH and VREFL pins.

The DAC module are connected to the APB bus and located at 0x40900000 The following are the I/O register for any of the ADC modules:

Register	Offset	Function
DACCTRL	0x0	DAC control register 0: DAC Enable 1: VREF Select - 0: External (VREFH, VREFL) 1: Internal (VDD, VSS)
DACDATA	0x4	0-9: Data

Analog Comparator

Raptor has one analog comparator that generates a digital output as a result of comparing two analog signals applied to its positive and negative inputs. The comparator is shown in the Figure 1 along with the relationship between the analog input levels and the digital output.

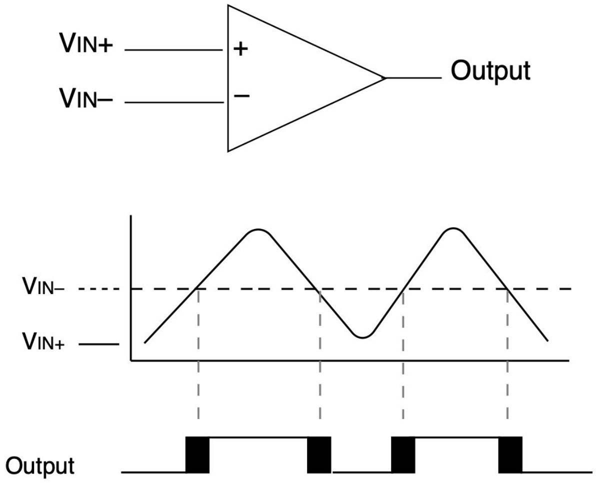
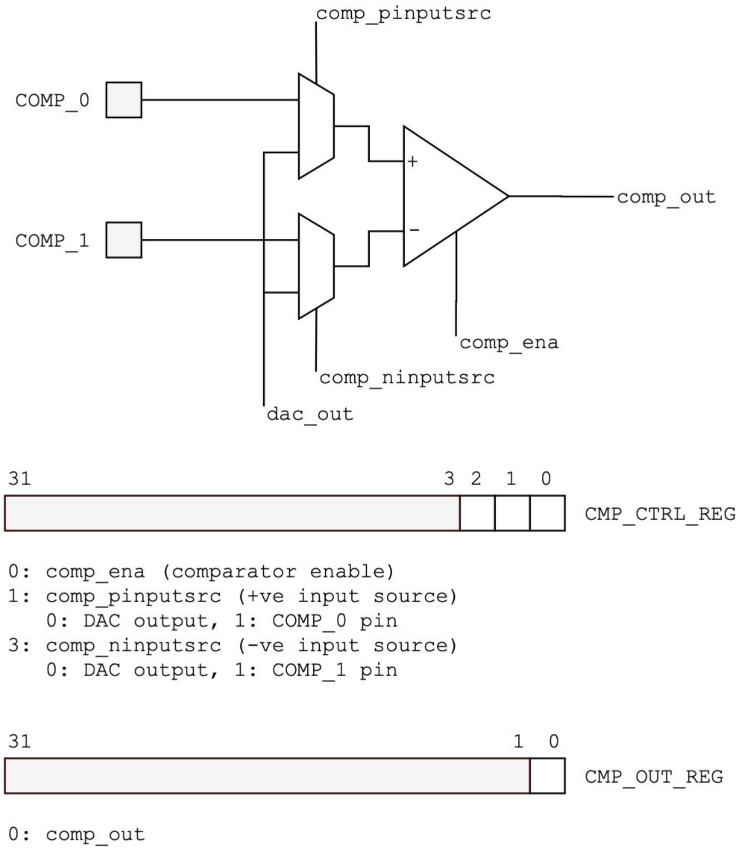


Figure 1. Comparator Operation

When the analog input at VIN+ is less than the analog input VIN–, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN–, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 1 represent the uncertainty due to input offsets and response time.

The analog comparator has two multiplexers at its inverting and non-inverting inputs to connect them to either to the on-chip digital to analog converter (DAC) output or to two dedicated external pins (COMP_0 and COMP_1) as shown in the Figure 2 below. The multiplexer selection lines can be controlled using the CMP_CTRL_REG (address: 0x40a00004) bits 1 and 2. Bit 0 or the same register can be used to enable or disable the comparator. Disable the comparator when it is not in use to save power. The output of the comparison can be read from Bit 0 of the register CMP_OUT_REG (address: 0x40a00000).



```
0: comp_ena (comparator enable)
1: comp_pinputsrc (+ve input source)
   0: DAC output, 1: COMP_0 pin
3: comp_ninputsrc (-ve input source)
   0: DAC output, 1: COMP_1 pin
```

```
0: comp_out
```

Figure 2. Analog Comparator Connections and Registers