Reference DESIGN

High-Performance 1024-Point Complex FFT

April 8, 1999 Application Note

XILINX

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
FAX: +1 408-559-7114
Email: coregen@xilinx.com
URL: http://www.xilinx.com/ipcenter

Features

- High-performance 1024-point complex FFT
- 16-bit complex input and output data
- 2's complement arithmetic
- Integrated direct memory controller (DMAC) to simplify host and memory interfacing
- High performance and density guaranteed through Relational Placed Macro (RPM) mapping and placement technology

1 Functional Description

The xFFT1024 fast Fourier transform (FFT) Core computes a 1024-point complex FFT. The input data is a vector of 1024 complex values represented as 16-bit 2's complement numbers – 16-bits for each of the real and imaginary component of a datum.

2 Theory of Operation

The discrete Fourier transform (DFT) $X(k), k = 0,\ldots, N-1$ of a sequence $x(n), n = 0,\ldots, N-1$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jnk2\pi/N} \qquad k = 0,\ldots, N-1 \tag{1}$$

where N is the transform size and $j = \sqrt{-1}.$ The *fast Fourier transform (FFT)* is a computationally efficient algorithm for computing a DFT.

The Xilinx 1024-point transform engine employs a Cooley-Tukey radix-4 decimation-in-frequency (DIF) FFT [1] to compute the DFT of a complex sequence. In general, this algorithm requires the calculation of columns or *ranks* of radix-4 butterflies. These radix-4 butterflies are sometimes referred to as *dragonflies.* Each processing rank consists of $N/4$ dragonflies. For $N = 1024$ there are 5 dragonfly ranks, with each rank comprising 256 dragonflies.

The FFT processor input-data for the Core is a vector of 1024 complex samples. The real and imaginary components of each sample are represented as 16-bit 2's complement numbers. The data is stored externally to the FPGA. An additional bank of scratchpad RAM is also required. The phase factors used in the FFT calculation are generated within the Core, so only the two

1024 × 32-bit banks of RAM discussed above are required to realize a complete transform processor. Like the input-data, the phase factors are kept to a precision of 16 bits.

All of the control signals required to interface the FFT module to external memory are generated by the Core. An integrated *direct memory controller (DMAC)* is also provided to allow the user to easily download data vectors for processing, and to read-back the previous transform result.

Two modes of operation are supported – *Continuous* and *One-shot.* The continuous mode of operation allows concurrent I/O and processing. As described above, 5 dragonfly ranks are performed to compute a 1024-point transform. When the Continuous mode of operation is selected, the host may download a new vector of data to the FFT engine while the 5th and final rank is being computed. Concurrent with the data load operation, the result of the current FFT is presented on the Core result databus. The FFT output samples are in *digit reversed order*, not *bit reversed order* as is the case with many conventional Cooley-Tukey radix-2 FFT algorithms.

3 Finite Word Length Considerations

3.1 Scaling

The radix-4 FFT algorithm processes an array of data by successive passes over the array. On each pass, the algorithm performs dragonflies, each dragonfly picking up four complex numbers and returning four complex numbers to the same addresses but in a different memory bank. The numbers returned to memory by the processor are larger than the numbers picked from memory. A strategy must be employed to accommodate this dynamic range expansion. A full explanation of scaling strategies and their implications is beyond the scope of this document, the reader is referred to several papers available in the open literature [2] [3] that discuss this topic.

The Xilinx 1024-point FFT Core scales dragonfly results by a factor of 4 on each processing pass. The scaling results in the final output sequence being modified by the factor $1/1024$. Formally, the output sequence $X^{'}(k), k = 0,\ldots,N-1$ computed by the Core is defined in Eq. (2)

$$X^{'}(k) = \frac{1}{1024} X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-jnk2\pi/N} \quad k = 0,\ldots,N-1 \tag{2}$$

4 Pinout

The schematic symbol is shown in Figure 1.

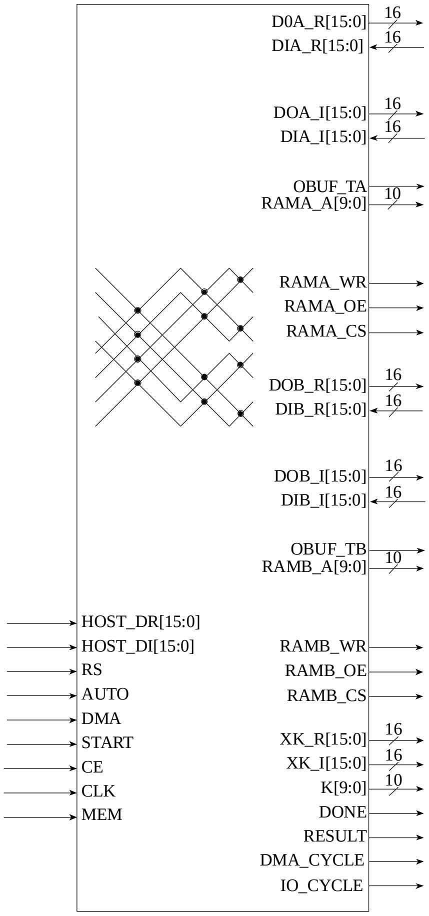


Figure 1: 1024-point FFT symbol.

Table 1 defines the module pin functionality.

5 Applications Information

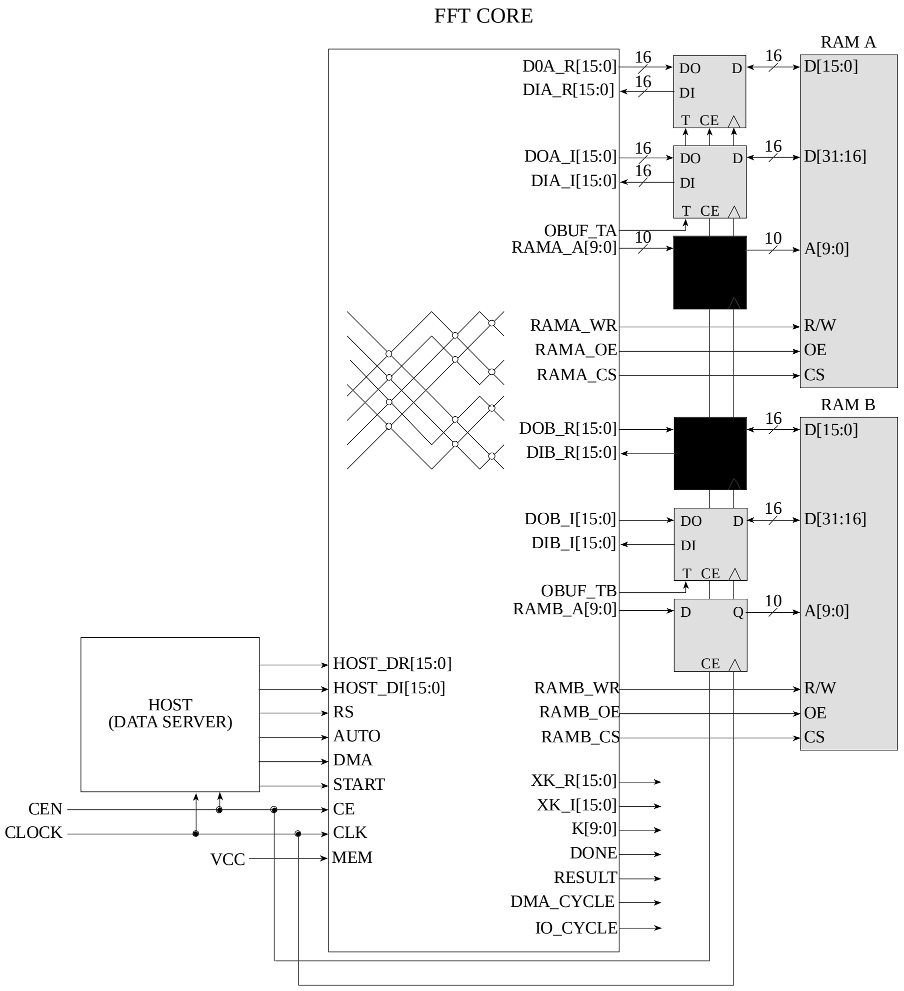
Figure 2 shows the recommended method for interfacing the FFT Core to external memory. The data buses DOA_[R|I], DOB_[R|I], DIA_[R|I], DIB_[R|I], RAMA_A and RAMB_A must be registered in the FPGA IOBs. Figure 3 shows how to register the memory data bus in the FPGA IOBs using *OFDEX16* and *IFDX16* library components. Figure 4 illustrates the use of *OFDX* elements to register the address bus.

Signal	Direction	Description	Signal	Direction	Description
RS	Input	Master reset – active high	RAMA_OE	Output	RAM A output enable – active low
START	Input	Initiate FFT execution – active high	RAMB_OE	Output	RAM B output enable – active low
AUTO	Input	Defines *One-shot* or *Continuous* mode	RAMA_CS	Output	RAM A chip select – active low
DMA	Input	Direct memory access request – active high	RAMB_CS	Output	RAM B chip select – active low
CE	Input	Clock enable – active high	RAMA_WR	Output	RAM A write strobe – active rising-edge
CLK	Input	Master clock input – active positive edge	RAMB_WR	Output	RAM B write strobe – active rising-edge
HOST_DR	Input	Host databus – real	OBUF_TA	Output	Tri-state control for IOB FFs
HOST_DI	Input	Host databus – imag.	OBUF_TB	Output	Tri-state control for IOB FFs
DOA_R	I/O	RAM block A databus - real	RESULT	Output	FFT result strobe – active high
DOA_I	I/O	RAM block A databus – imag.	XK_R	Output	FFT result bus - real
DOB_R	I/O	RAM block B databus - real	XK_I	Output	FFT result bus – imag.
DOB_I	I/O	RAM block B databus – imag.	K	Output	FFT result index bus
RAMA_A	Output	RAM A address bus	DONE	Output	FFT complete strobe – active high[1]
RAMB_A	Output	RAM B address bus	DAM_CYCLE	Output	Host DMA in progress strobe – active high
MEM	Input	Defines single or dual address-space operation	IO_CYCLE	Output	Precedes activation of *STROBE* by 4 clock cycles and indicates a pending I/O operation[2]

NOTES:

1. *DONE* is active for one clock period.
2. *IO_CYCLE* is active for one clock period.

Table 1: 1024-point FFT pin definitions.



CEN = SYSTEM CLOCK ENABLE
CLOCK = SYSTEM CLOCK

Figure 2: FFT memory interfacing – dual memory address space operation.

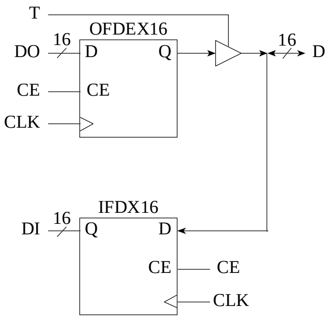


Figure 3: Recommended method for registering the memory data bus in the FPGA IOBs.

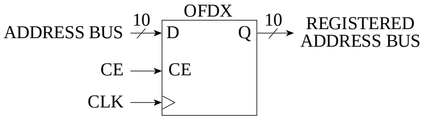


Figure 4: Recommended method for registering the memory address bus in the FPGA IOBs.

6 Timing and Control

The FFT core supports two modes of operation: *Continuous* and *One-shot.* The mode of operation is selected by the *AUTO* input signal.

6.1 Continuous Mode

With the *AUTO* control signal high, the continuous mode of operation is selected. In this mode, transforms can be computed in a block-continuous fashion, with I/O operations performed concurrently and effectively invisible to the transform engine. The typical sequence of events for using this mode are

1. Use the Core DMAC to load the first block of data into memory
2. Initiate transform execution by asserting the *START* signal high for one clock period
3. Supply new data to the Core over the host databus when the *RESULT* strobe goes active.
4. Read the transform data from the result bus XK_[R|I] when *RESULT* is active – the index bus *K* supplies the frequency index for the transform output samples

The host downloads a new vector of data to the Core while the final dragonfly rank is being performed. During the final rank, the FFT result vector is presented on the *XK*_[*R*|*I*] data bus*.* The *RESULT* strobe indicates valid transform data. As indicated in Figure 5, the result data is *out-of order*. The index bus *K* identifies the DFT samples. The *K*-bus can be used as memory address bus to a block of RAM that is used to simultaneously store and re-order the output samples.

The *DONE* signal identifies the end of the result window and becomes active one clock cycle prior to the last DFT result, *X*(1023), is present on the result bus. The *RESULT* strobe can be used as a clock enable for an FFT post-processor, result FIFO, or output register that is connected to a host processor.

When the Continuous mode of operation is used, the *START* signal is asserted only once to initiate the first transform. The processor will continue to perform FFTs back-to-back once it has been started in continuous mode. The user must not assert *START* again once processing has started in this mode.

6.1.1 Performance – Continuous Mode

In continuous mode, I/O and computation are overlapped. As stated above, while the final processing stage is in progress, the host can download a new data vector. There are 5206 clock cycles from the time *START* is asserted to the time the last DFT value appears on the result databus. During the final 1024 clock cycles, the host is downloading a new input vector. The transform execution time T_{C} is

$$T_{\mathrm{C}} = \frac{5206}{f_{\mathrm{CLK}}}$$

where f_{CLK} is the system clock frequency. Execution times for several clock frequencies are shown in Table 2.

Clock Frequency	52 MHz	58 MHz	60 MHz	68 MHz
FFT Execution Time	100.1 µs	89.9 µs	86.8 µs	76.6 µs

Table 2: FFT execution times for several values of system clock frequency.

6.2 Output Data Ordering

As described above, the result data is computed in a digit-permuted manner. For a 1024-point data-set, $\log_2 1024 = 10$ bits of address are required to reference the samples. Let the digit positions in an address word *A* be denoted by $\{a_9, a_8, \ldots, a_0\}$, $a_i \in \{1,0\}$ for $i = 0, \ldots, 9$. a_9 is the most-significant digit while a_0 represents the least-significant bit. The familiar bit-reversed permutation associated with many common Cooley-Tukey FFT algorithms is defined by the bit permutation defined in Eq. (3)

$$\{a_0, a_1, \ldots, a_9\} \quad a_i \in \{1,0\},\ i = 0, \ldots 9 \tag{3}$$

For the radix-4 architecture employed by the FFT Core, the output indexing is described by the following permutation

$$\{a_1, a_0, a_3, a_2, a_5, a_4, a_7, a_6, a_9, a_8\} \quad a_i \in \{1,0\},\ i = 0, \ldots, 9 \tag{4}$$

6.3 One-Shot Mode

One-shot mode is selected when the *AUTO* input pin is low. The typical sequence of events for using this mode is to

1. Use the CORE DMA controller to load the input vector into memory *A*, while simultaneously reading back the result of the previous transform from memory bank *B.*
2. Initiate transform execution by asserting *START* high for one clock cycle.

Unlike Continuous mode, the *START* signal must be asserted to initiate each new calculation.

6.3.1 Performance – One-Shot Mode

One-shot mode differs from the *continuous* mode in one major way. *One-shot* mode involves an additional memory access phase for I/O. The transform time will be the same as for *Continuous* mode, but an additional 1024 + 1 clock cycles are required for I/O. The additional `+1' accounts for the time from the final result value being produced by the Core, latched in the FPGA IOBs, and written to external memory. In One-shot mode, the FFT engine re-orders the DFT samples as they are written to memory during the final processing stage. Both memory banks are required for all 5 processing stages, and the user must wait until the final DFT value has been written to memory before a new DMA cycle can be initiated. The DMA cycle will load new data for processing, while simultaneously reading back the results of the previous calculation.

If the host cannot support concurrent Core write and read accesses, two DMA operations will be required – one to read the DFT samples and a second to write a new data vector to memory. *The read operation must occur first in this sequence of operations.*

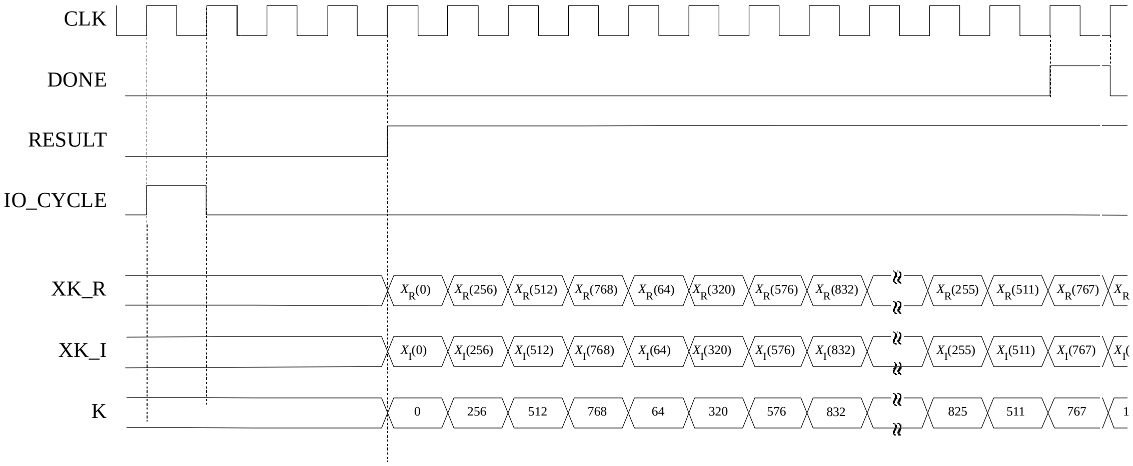


Figure 5: Result strobe timing.

Just like the *Continuous* operation mode, the DFT samples are available on the result bus *XK_[R|I].* In *One-Shot* mode the DFT values are presented on the result bus in natural order – recall that this is different to *Continuous* mode. Since the result is available in natural order, the index bus *K* should not be used at all[1]. The timing for writing data to the Core and reading the result of the previous operation is shown in Figure 6. Note that the DFT values are available in-natural order starting with the DC term *X*(0) followed by samples *X*(1), *X*(2) and so on up to the lowest negative frequency component *X*(1023).

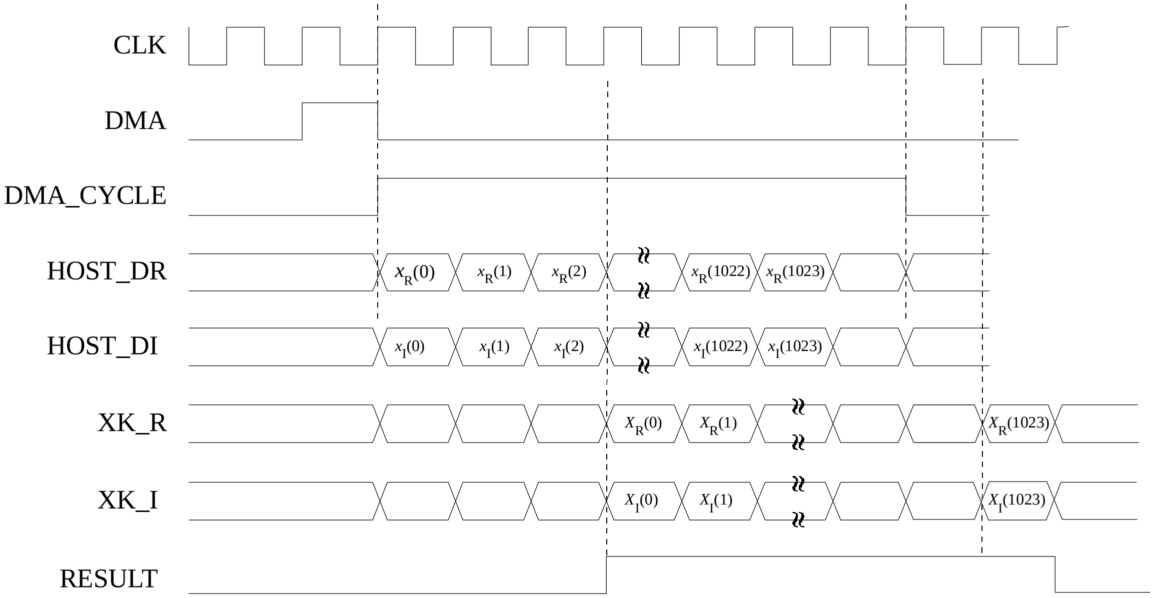


Figure 6: Host DMA timing.

6.4 START and Control Signal Timing

Figure 9 shows the relationship between the *START* signal and the Core/memory control signals, as well as the RAM[A|B] address buses. Asserting *START* high for a minimum of 1 clock period initiates FFT execution. *START* does not have to be synchronized with the master clock, but the signal must be high for at least one positive-going transition of *CLK.*

6.5 IO Pending FLAG - IO_CYCLE

The *IO_CYCLE* output signal provides advance information of an I/O operation. This control signal is most useful when the Core is operated in continuous mode. Recall that in this mode of operation, the host data-server must present new data to the FFT processor during the final processing phase. This time period is framed by the *RESULT* strobe. As shown in Figure 5 *IO_CYCLE* becomes active 4 clock cycles prior to the assertion of the *RESULT* strobe. Using this signal allows the host data server to take appropriate action should the situation arise where the Core timing requirement for new data cannot be met. For example, the data source may be temporarily stalled. To avoid having the entire system fail, the user may choose to temporarily remove the clock enable signal *CE* until new input data is available.

[1] The *K* index bus is only provided for un-scrambling the DFT samples when they are presented by the Core in digit-permuted order. This only occurs when *Continuous* operation is used OR when the Core is operated with a single memory address space – refer to Section 7.

Note, placing *CE* in the in-active state will stop all Core activity. This includes external memory accesses as well as activity on the result databus.

6.6 Clock-Enable – *CE*

There are several issues involving the clock-enable *CE* pin that designers should be familiar with when developing systems with this core. *CE* is a high fan-out signal and should be presented to the Core via a low-skew clock buffer, either *BUFGP* or *BUFGS,* to achieve maximum operating frequency*.* Refer to the Xilinx 4000 series device data book for more information on these features. **The *CE* pin is a master clock enable for the entire Core**. When in the inactive state, all core operations are stalled until *CE* is re-asserted.

6.7 System Reset - *RS*

The FFT processor can be reset at any time by asserting *RS* high for at least one clock cycle. This will stop all Core activity including external memory accesses. Any transform currently in progress will not be completed and the external memory is not guaranteed to contain any valid transform results.

7 Using the FFT Core with a Single Address Space

Figure 7 shows how the Core can be used with a single address space. The *MEM* pin must be connected to logic 0 and the a new clock, *CLK1*, that is half the frequency of the primary clock, *CLK*, must be generated and supplied to the module clock pin. The figure shows the recommended method for generating the half-rate clock using a T-flip-flop. Since the FFT clock net has a very high fan-out, *CLK1* should be presented to the core via a low-skew clock buffer, preferably *BUFGP*. The additional address and data registers **must** be connected as shown. Note that *CLK1* is used as a clock enable for these registers.

A slightly different memory interface to that used for dual memory space operation is required when using a single address space. Separate clock enables for the data input and output IOB latches. These are referred to as *CEI* and output *CEO* in the IOB interface circuits labeled *IOB_REG* in Figure 7. Figure 8 shows the details of the recommended memory interface circuit.

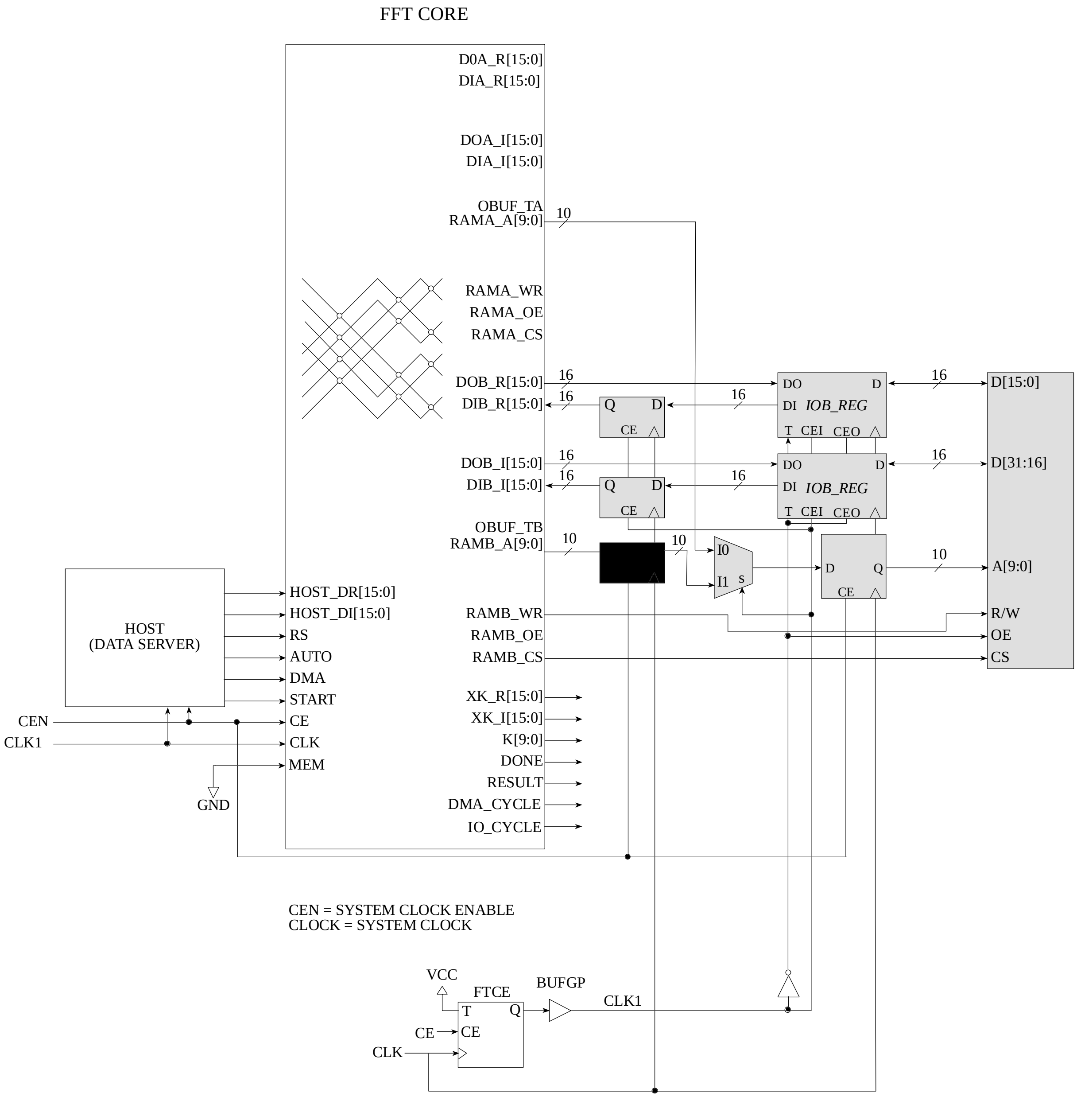
The connection diagram in Figure 7 does not show the system master clock enable *CEN* operating in conjuction with the two *IOB_REG* components. If this signal is to be used to suspend execution it must be combined with the signals that generate *CEI* and *CEO*. This can be accomplished with two AND gates.

When using the single memory address space option the transform result will always be presented on the output bus in digit reversed order. The index bus must be used to re-order the result if a naturally ordered dataset is required.

The *MEM* signal must be connected to either a logic 0 or logic 1, it cannot be connected to an input pad.

7.1 Performance – Single Address Space Operation

When a single external address space is employed with the FFT module the execution time, in comparison to dual memory space operation, is halved. The external memory interface is still clocked at the same frequency as dual memory operation, but the Core clock frequency is half of its dual memory rate. Now a single memory port must support all Core data accesses, with reads and writes occurring on alternating edges of the high frequency clock *CLK* in Figure 7*.*



CEN = SYSTEM CLOCK ENABLE
CLOCK = SYSTEM CLOCK

Figure 7:Using the FFT Core with a single address space. The *MEM* pin must be connected to logic 0.

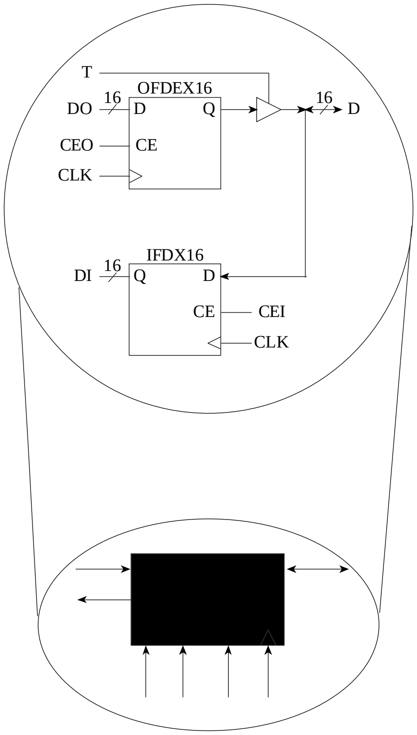


Figure 8: Memory interface for FFT Core - single memory address space.

8 Core Resource Utilization

The number of CLBs used for this module is slightly different depending on the mode selected. In continuous mode the Core uses 1495 CLBs. In one-shot mode, 1494 CLBs are required. The geometry of the RPM requires it to be placed in a XC4062XL or larger device.

9 Ordering Information

This macro is provided free of charge to all Xilinx customers. For additional information contact your local Xilinx sales representative, or e-mail requests to dsp@xilinx.com.

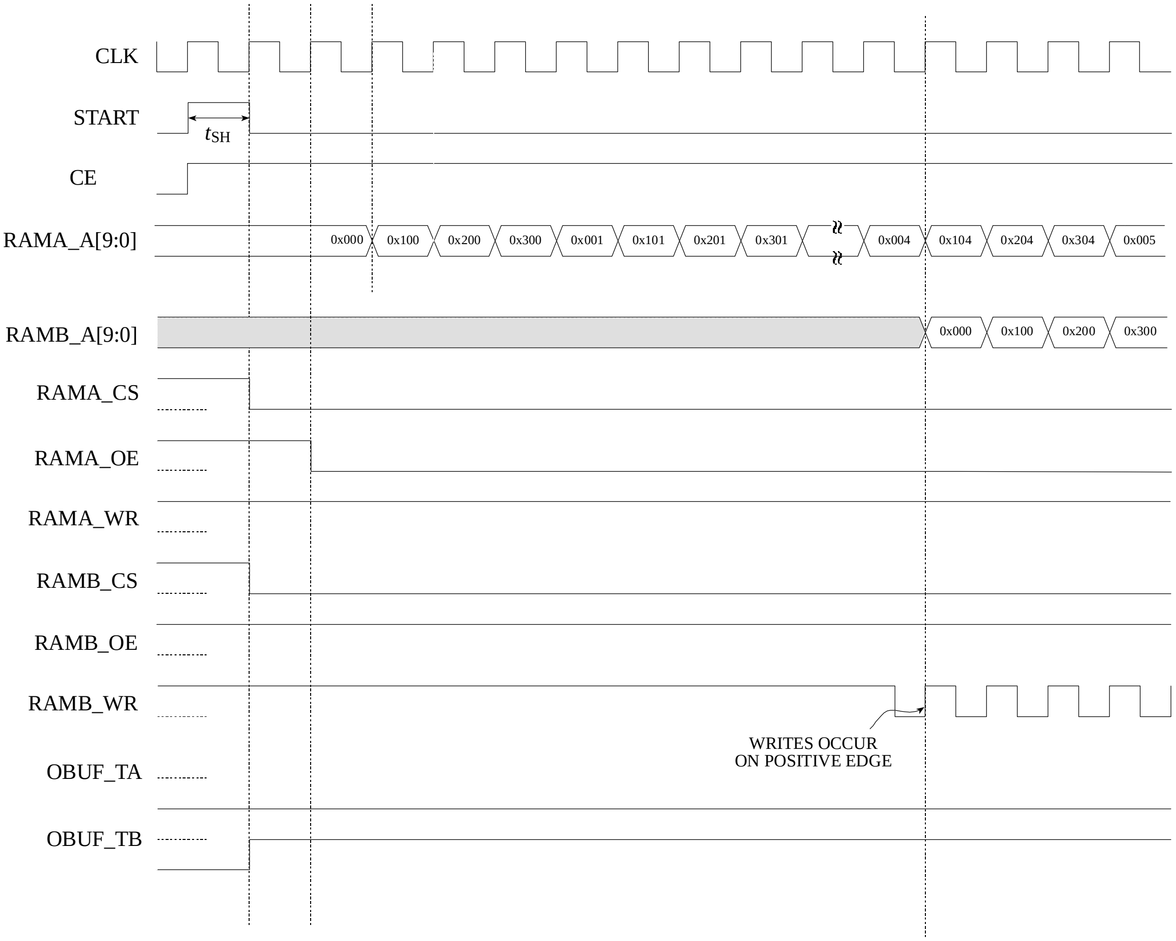


Figure 9: *START* and control signal timing.

10 References

[1] J. W. Cooley and J. W. Tukey, ``An Algorithm for the Machine Calculation of Complex Fourier Series'',
Math. Comput., Vol. 10, pp. 297-301, April 1965.

[2] W. R. Knight and R. Kaiser, ``A Simple Fixed-Point Error Bound for the Fast Fourier Transform'', *IEEE Trans. Acoustics, Speech and Signal Proc.,* Vol. 27, No. 6, pp. 615-620, Dec. 1979.

[3] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing,* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1975.

Xilinx Reference Design License

By using the accompanying Xilinx, Inc. Reference Designs (the “Designs”), you agree to the following terms and conditions. You may use the Designs solely in support of your use in developing designs for Xilinx programmable logic devices or Xilinx HardWire™ devices. Access to the Designs is provided only to purchasers of Xilinx programmable logic devices or Xilinx HardWire™ devices for the purposes set forth herein.

The Designs are provided by Xilinx solely for your reference, for use as-is or as a template to make your own working designs. The Designs may be incomplete, and Xilinx does not warrant that the Designs are completed, tested, or will work on their own without revisions. The success of any designs you complete using the Designs as a starting point is wholly dependent on your design efforts. As provided, Xilinx does not warrant that the Designs will provide any given functionality, and all verification must be completed by the customer.

Xilinx specifically disclaims any obligations for technical support and bug fixes, as well as any liability with respect to the Designs, and no contractual obligations are formed either directly or indirectly by use of the Designs. XILINX SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION DIRECT, INDIRECT, INCIDENTAL, SPECIAL, RELIANCE OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF THE DESIGNS, EVEN IF XILINX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Xilinx makes no representation that the Designs will provide the functionality you are looking for, or that they are appropriate for any given use. Xilinx does not warrant that the Designs are error-free, nor does Xilinx make any other representations or warranties, whether express or implied, including without limitation implied warranties of merchantability or fitness for a particular purpose. The Designs are not covered by any other license or agreement you may have with Xilinx.

The Designs are the copyrighted, confidential and proprietary information of Xilinx. You may not disclose, reproduce, transmit or otherwise copy the Designs by any means for any purpose not set forth in this license, without the prior written permission of Xilinx.

You agree that you will comply with all applicable governmental export rules and regulations, and that you will not export or reexport the Designs in any form without the appropriate government licenses.

XILINX, INC., 2100 Logic Drive, San Jose, California 95124