

Chapter 9

Detailed Low Level Design and Functional Testing

This chapter deals with the low level logic design of the four operations implemented with FPGAs: the FFT, frequency manipulation block, IFFT and the time manipulation block. Circuitry to implement all the required functionality is developed including memory addressing and clock division with the exception of start-up and SPROM initialisation circuitry. Where possible, logic functionality is checked and post device place and route performance determined. All designs were completed, verified and timed using the Xilinx Foundation 1.5i and Core Generator software. In addition some consideration is given to the required memory interface chips with suitable components identified.

The designs presented are in no way the only viable solution. Indeed experienced FPGA logic designers would be expected to arrive at considerably different and better solutions. Instead they should be viewed as an initial starting point for further development.

9.1 FFT FPGA

This FPGA performs a FFT on the 1024 time samples stored in the ADC – FFT memory interface each time sampling window. The 1024 DFT frequency sample results are written to the FFT – frequency

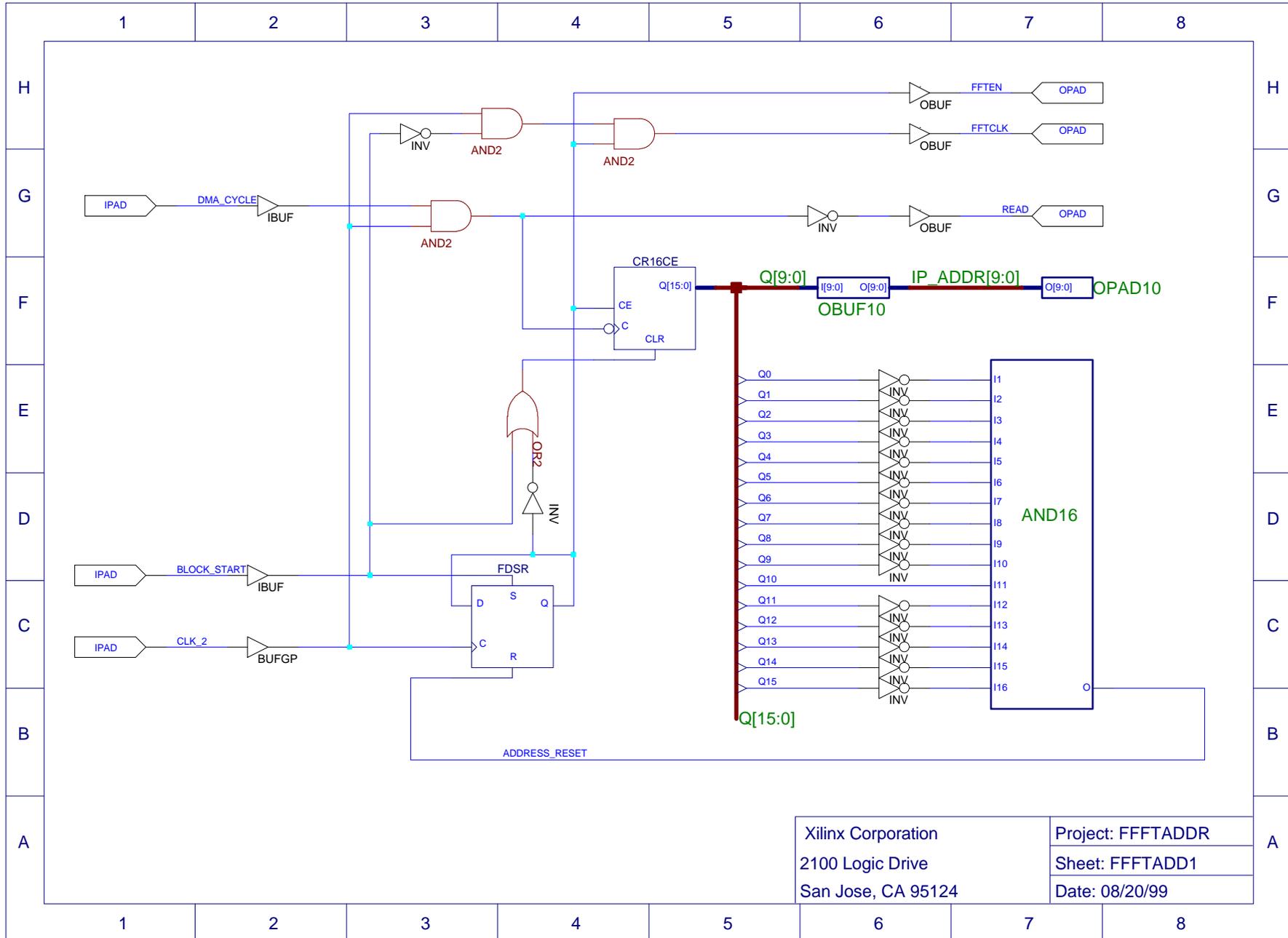


Figure 9.1 FFT additional control circuitry for real time data input from the ADC - FFT memory interface

manipulation FPGA memory interface.

All FFT FPGA schematic, simulation and associated files are located in the `:\Designs\Xilinx\Fft\fftref\fftaddr` and `\fftmem` subdirectories on the enclosed CD ROM.

9.1.1 Peripheral Circuit Requirements

As mentioned in chapter 8, the FFT Reference Design doesn't actively address its input memory. Logic is required to provide a read strobe and the address bus to the memory interface between the ADC and FFT, which serves the same purpose as the host data server shown in the Reference Design data sheet (on CD ROM). Page selection for the memory interfaces is external to the FFT FPGA design.

9.1.2 Logic Design

The FFT Reference Design produces a strobe called `DMA_CYCLE` when new input data is required during the last Butterfly rank process, which lasts for the duration of the transfer then goes low. This signal can simply be gated with the `FFTCLK` to produce a read strobe, which will be active only when the `DMA_CYCLE` is asserted high by the FFT. The rising edge of the `DMA_CYCLE` signal can also initiate a binary counter for the address bus, which is reset on a count of 1024. Figure 9.1 shows circuitry to produce the necessary address bus and control signaling for time sample input data from the ADC – FFT memory interface.

In addition to the input control timing circuitry, the FFT block requires its scratchpad RAM data and address buses to be synchronously registered. The data buses must also be converted to bi-directional operation for interfacing to external tri-state RAM. Figure 9.2 showing the complete FFT FPGA design, includes two macros `TRI_BUFF_32` and `OFDX10` to achieve this.

9.1.3 FFT FPGA Design Verification

After the problems encountered with the Core FFT, it seemed wise to verify the operation of the Reference Design before finalising its use within the simulator. Operational testing can be carried out in three parts:

- Firstly, verify that the control signal timing corresponds to that shown on the published data sheet.
- Secondly, verify that the FFT transform function is correct through the transform of a 1024 point real input data vectors and comparing the results with those generated for the same input vectors by a numerical package such as Matlab.
- Thirdly, verify timing and addressing for the input RAM circuitry.

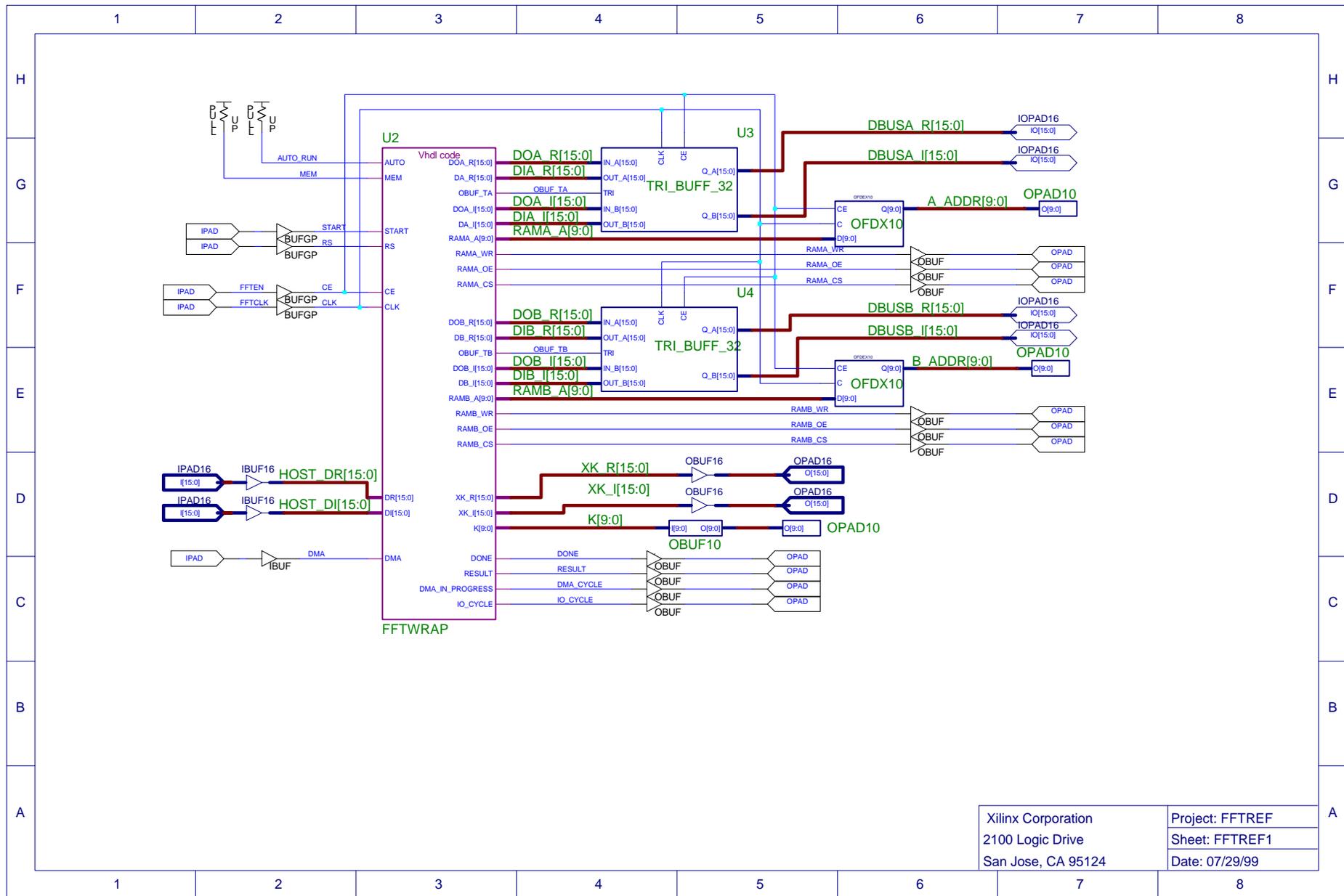


Figure 9.2 Complete FFT transform engine design

Unfortunately, the FFT Reference Design doesn't produce a behavioural model for the function, so only gate level simulation after the generation of a supplementary EDIF file is possible. This approach is extremely time consuming due to the very large design. From circuit modification to completion of a basic single 1024 point FFT simulation without scratchpad RAM, a processing time in excess of four hours is encountered using a PII 300 MHz processor. In addition, no suitable memory blocks of 1024 x 32-bit RAM for scratchpad use don't exist in Xilinx libraries, so FFT transform functionality can't be cleanly tested without custom design of suitable memory blocks for inclusion in a schematic design. Because of the large size of such memory blocks (2 x 1024 by 32-bits), the simulation time would be enormous. An alternative to inserting scratchpad RAM into the schematic design is to verify the FFT transform through a complex simulation command file which mimics the behaviour of the RAM. The command file must record temporary process data output from the FFT Butterfly ranks on complex data buses DBUSA and DBUSB (figure 9.2), then re-inject it into the FFT engine as the scratchpad RAM would do when addressed by the FFT engine in an actual physical circuit. There is no easy way to do this with the Xilinx Foundation software other than by observing the address and data buses of the scratchpad RAM through each Butterfly rank process and then copying the values into the command file for re-injection to the FFT engine in the following rank process, carefully noting the read/write ordering from the address buses. This is extremely laborious and took over a day to write the command file required to test a single complete 1024 point FFT transform.

9.1.3.1 FFT Reference Design Control Timing

Control timing verification is by far the easier of the three verification steps. The command file `fftref_control_verification.cmd` in the `fftref` directory on the CD contains a command file written to simulate an initial data vector load into the FFT engine, then a complete 5206 clock cycle 1024 point, followed by another 5206 cycles to verify the repetitive nature of the control timing for a second FFT operation. The simulation can be re-run or the results viewed from the file `fft_ct_v.tve` as simulation file takes about an hour to complete. Close examination of the control strobes and address buses confirm the published control timing and addressing.

9.1.3.2 FFT Reference Design Discrete Fourier Transform

As previously mentioned, transform functional verification is not easy in the Foundation design environment. The first method of including scratchpad RAM into a design schematic with the FFT Reference Design was initially attempted. Custom 1024 by 32-bit RAM was constructed from primitive library components and is included in the `fttmem` subdirectory on the CD, but is far too large and numerous to print! Unfortunately, this approach wasn't successful due to compilation faults when running the simulation software – this may be due to the shear size of the combined FFT block and memory as the Foundation software is written for designs which can fit into a single FPGA.

The second method of writing a complex simulation command file to mimic the action of the scratchpad RAM was reluctantly embarked upon. Due to the time consuming nature of this approach, only one transform has been tested; that of a simple sampled sinusoidal signal. Ideally the transform for

many different input sample vectors should be simulated and each result compared with that generated by Matlab for the same input vector before the functional integrity of the transform can be assured.

The simulation command file `fftfref_transform_verification.cmd` contains the script to simulate the FFT on a sampled sinusoid, with the scratchpad RAM data written by the FFT engine during the transform stored in the files `A_BR1.dat`, `B_BR2.dat`, `A_BR3.dat`, `B_BR4.dat` and `A_BR5.dat`. The associated waveforms are stored in the file `fft_tf_v.tve`. Again, the simulation takes about an hour to complete. The Excel file `fftfref_transform_verification.xls` tabulates the data written to the scratchpad RAM by the FFT engine during the five Butterfly rank processes in the order in which it is read back into the FFT, as well as the final transform result.

The simulation is for the FFT of a real set of 1024 time samples from a sinusoid with 128 cycles within the 1024 point sampling window, of the form

$$x(kT_s) = [0, 0.7071, 1, 0.7071, 0, -0.7071, -1, -0.7071, 0, 0.7071, 1, 0.7071, 0, -0.7071, -1, -0.7071 \dots]$$

where the sinusoid time samples shown repeat 128 times within the time sampling window. The DFT of such a signal should be purely imaginary with just two non-zero components of $0.5i$ at frequency sample $+128$ and $-0.5i$ at sample 896 (equivalent to -128 with a 1024 point transform). **Examination of the FFT output vectors on data buses XK_R and XK_I from the simulation waveform shows that the FFT Reference Design's result doesn't match the correct values for the DFT of an input vector describing the sampled sinusoid.** Instead the FFT Reference Design seems to produce a set of 32 non-zero imaginary frequency samples of $+0.5i$ at frequency samples 192 through 207 and $-0.5i$ at samples 384 through 399. This is somewhat difficult to explain. The result, which contains only imaginary components, is consistent with the correct DFT for a real, odd time function such as the sine wave. Even the nature of the imaginary result seems indicative of a DFT having occurred (only two imaginary magnitudes of $0.5i$ and $0i$), but the placing and sixteen times replication of the non-zero samples are incorrect.

Contact with the Design's author, Dr. Chris Dick of Xilinx, hasn't resolved the problem. It seems he hasn't done a complete simulation using the Foundation tools for the reasons iterated concerning scratchpad memory requirements and behavioural models. Dr. Dick has performed a transform using a Viewlogic simulation tool called 'Viewsim', unavailable to myself. A copy of the e-mail from Dr. Dick is included in appendix 7 and worth reading. The author suggests a new FFT design targeted at the Virtex series of devices instead of continuing with the XC4000 Reference Design. The Virtex FPGAs really are enormous, with upto 6 144 CLBs and 130 kbits of RAM compared to the largest XC4085XLA device containing 3 316 CLBs and no additional RAM. Virtex gate counts exceed 1 million compared to just over 100 000 for the XC4082XLA. The use of Virtex devices would solve all the simulation problems as a single design could also include scratchpad RAM. Unfortunately, the Virtex FFT design hasn't been released to date, but potentially provides a good transform solution.

9.1.3.3 Input RAM Addressing and Control

The command file `fft_input_address_verification.cmd` in the `fftaddr` subdirectory

simulates two complete time sampling windows. The address and timing waveforms may be viewed are stored from the file `fftaddr.tve`.

9.1.4 FFT Performance

Although actual transform functional verification hasn't been achieved, performance has been evaluated both before and after FPGA device placement and routing. Pre-routing performance is determined from the logic delay and estimated path delays within the design. Post routing performance includes the actual net delays encountered when the design has actually been placed within a target FPGA. Using the fastest XC4082XLA-07HQ240 device the maximum FFT clock frequencies are:

Timing Analysis	Maximum Clock Frequency
Pre place and route	69 MHz
Post place and route	42.3 MHz

Table 9.1 Pre and post place and route FFT performance

The large difference in pre and post performance is due to the absence of a User Constraints File (UCF) during the Foundation place and route operation. UCF files tell the Foundation design compiler which paths need to be timed 'together', which can be ignored and which are most critical. It is somewhat surprising that no UCF file is supplied with the FFT Reference Design. Because the design is only shipped as overall FFT macro and doesn't include the basic files (the net level files) from which the overall block is constructed, it is difficult to write a UCF file without guessing which constraints to apply to the paths identified in the Foundation timing reports. However, with trial and error, the post place and route clock frequency should approach the limit of 69 MHz. Even at 42 MHz, the complete 1024 point FFT execution time is just 125 μ s (5205 x 24 ns), clearly within the 232 μ s time sampling window. If future development for VDSL line simulation use the FFT Reference Design, an optimal UCF to maximise the FFT clock frequency at 69 MHz will be required giving an execution time of 75 μ s.

9.1.5 IFFT FPGA

As the functional integrity of the FFT Reference Design hasn't been proved, the design of the IFFT module has been left to further work. The internal structure of the IFFT logic will be very similar to that for the FFT module taking into account appropriate FFT / IFFT scaling.

9.2 Frequency Manipulation FPGA

Each time sampling window, the frequency manipulation FPGA multiplies the 1024 16-bit complex DFT frequency samples with 1024 16-bit complex coefficients then adds the result to another 1024 16-bit complex numbers, downloaded from the controlling PC in the previous sampling window.

All schematic, simulation and other files for the frequency manipulation FPGA are located in the `:\Designs\Xilinx\frqmanip` directory on the enclosed CD.

9.2.1 Peripheral Circuit Requirements

As shown in figures 8.3 and 8.4, the 1024 DFT frequency samples are stored in sequential order in paged dual port RAM. The complex samples are separated into real and imaginary parts, each complex component occupying 16-bits of a 32-bit wide memory word. The manipulation block reads and modifies these samples then writes them to its output memory interface. The manipulation block must provide addressing for the input and output RAM from which the frequency samples are read from and written to. In addition, the block must address the PC memory interfaces holding the multiplication and addition vectors downloaded in the previous sampling window from the PC. Memory page selection is external to the manipulation block.

The three memory interfaces containing input data for the frequency manipulation process must be sequentially addressed and strobed with a read pulse. The memory interface to which the results are to be written must be similarly addressed and provided with a suitably timed write strobe after each manipulation result is available. Figure 9.3 shows the address and read/write strobe timing for frequency manipulation process.

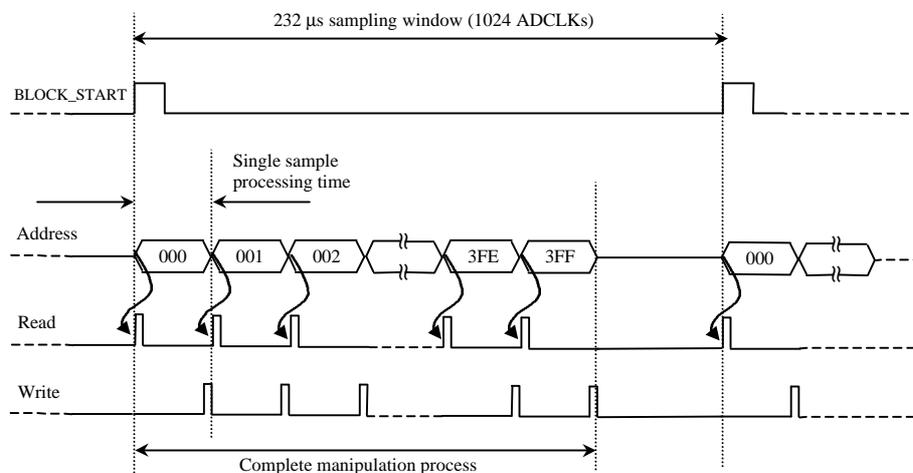


Figure 9.3 Frequency manipulation addressing and read / write strobe

9.2.2 Logic Design

9.2.2.1 Arithmetic Functions

The two arithmetic functions of addition and multiplication can be implemented using the Cores described in appendix 5. Two sets of 16-bit operation can be combined in parallel to allow the separate real and imaginary complex components to be manipulated at the same time, shown in figure 9.4. The eight data buses are as follows:

- A_LO – Real DFT samples (input)
- B_LO – Real multiplying coefficients (input)
- C_LO – Real addition components (input)
- A_HI – Imaginary DFT samples (input)
- B_HI – Imaginary multiplying coefficients (input)
- C_HI – Imaginary addition components (input)
- RESULT_LO – Manipulated real DFT sample (output)
- RESULT_HI – Manipulated imaginary DFT sample (output)

Attenuation is achieved through the method of ignoring the 16 LSBs of the result of the 16-bit multiplication of A and B. No account of carries produced by the addition process is included because this should never actually occur in practical use.

Each multiplication requires five clock cycles, with additions just one. Therefore, the result write strobe must occur at least six cycles after the read strobe.

9.2.2.2 Timing Functions

Each sample manipulation requires at least six clock cycles to perform. However, if eight cycles are allocated to each operation, the read and write strobe signals are easily derived by dividing the manipulation clock by eight (2^3). Each new 1024 sample frequency manipulation process is initiated by the assertion of the global BLOCK_START timing strobe, figure 9.3. After this strobe, the manipulation timing logic must produce 8×1024 (8192) clock cycles to complete the operation before the start of the next sampling window.

The required number of clock cycles are produced by gating the external clock with a control signal which is set high on the falling edge of the BLOCK_START signal and reset low after the 8192 cycles have been completed. A simple 16-bit library binary counter is used to count the number of cycles, with bits 4 to 13 acting as the address bus (effectively the clock divided by eight and counted). A 16-bit AND gate is used to generate the ADDRESS_RESET pulse after 8192 cycles have occurred. The read strobe is produced at the start of every eight clock cycles simply by dividing the manipulation clock by eight, and the write edge derived from the output from the third bit of the binary counter. Figure 9.5 shows logic to produce these timing signals.

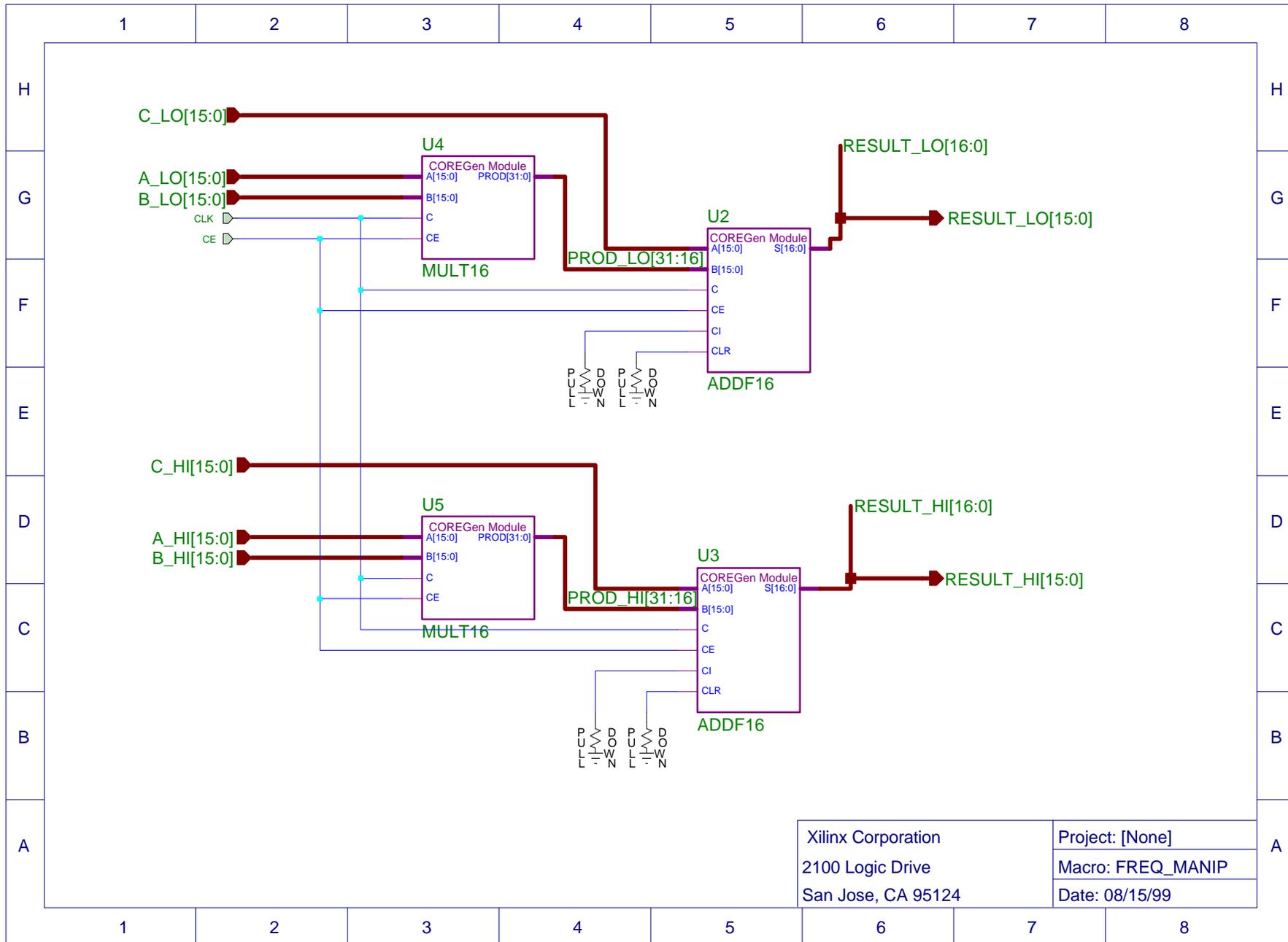


Figure 9.4 Frequency manipulation arithmetic functions

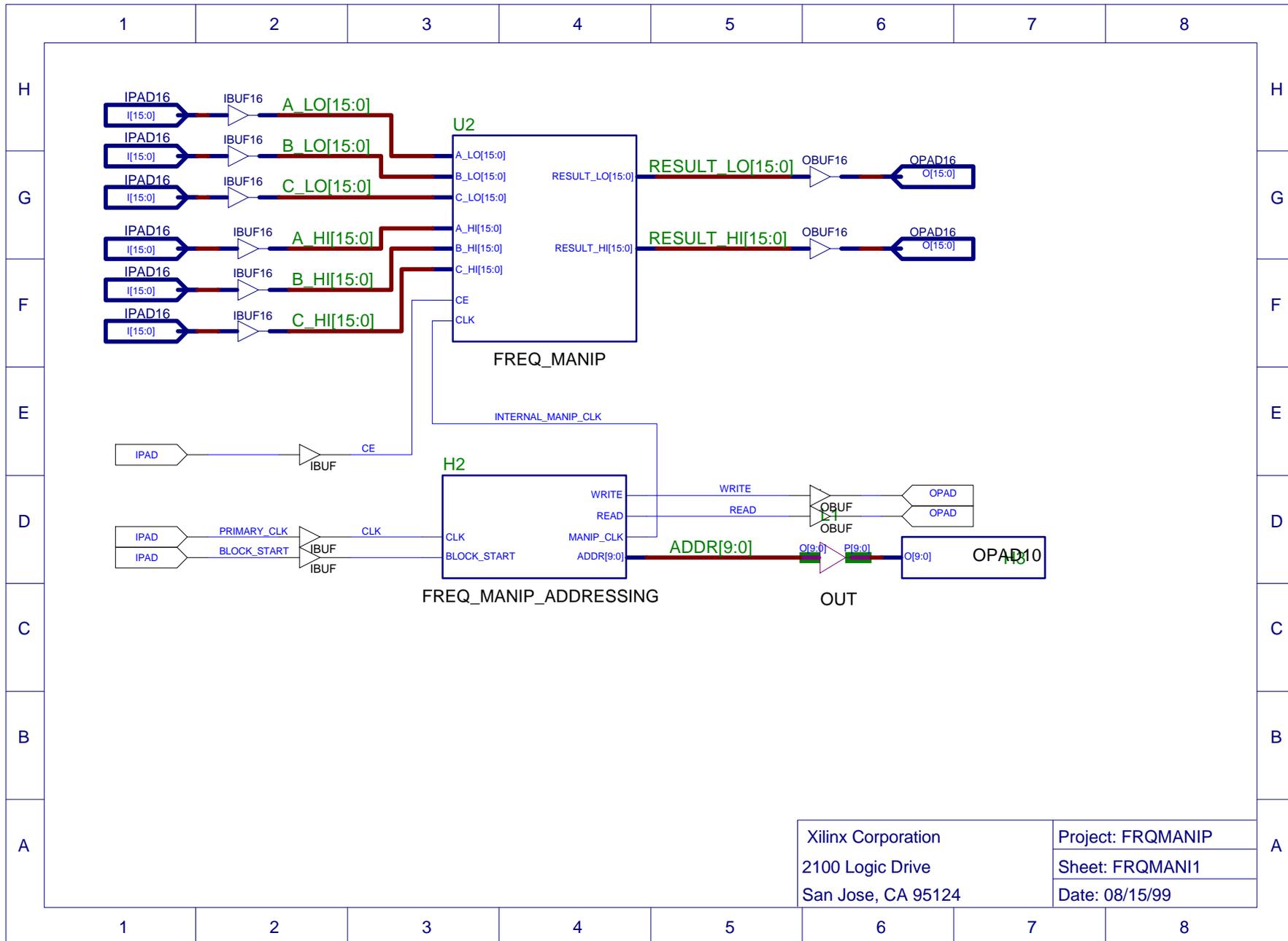


Figure 9.6 Frequency manipulation complete circuit

Figure 9.6 shows the combination of arithmetic and timing functions into a single unit.

9.2.3 Functionality Testing and FPGA Placement

Complete functionality testing has been undertaken on the overall frequency manipulation block. On the CD there is a command test file `frqmanip_verification.cmd` which can either be re-run, or the result's waveform viewed directly from the file `frqmanip.tve`. The waveform file shows the exact read/write and address timing.

The complete frequency manipulation block can be placed in the smallest XC4013XLA device and uses only 13% of the CLBs. This would allow the time manipulation function to also be placed within the same physical FPGA to reduce costs. In addition, other board logic can easily be placed in the same device as long as one with enough I/O pins is chosen.

9.2.4 Performance

Using the XLA-07 speed grade, post placement timing analysis suggests a maximum clock frequency of 125 MHz, in which case each manipulation of 1024 samples would be completed in approximately 65 μ s. Bearing in mind the external clock generation of all components, a manipulation clock at 70.656 MHz ($2^2 \times 17.664$) will give a complete process execution time of just 116 μ s, well within the 232 μ s the sampling window limit.

9.3 Time Manipulation FPGA

Each time sampling window, the time manipulation operation adds 1024 real numbers downloaded from the PC during the previous sampling window and stored in one of the PC memory interfaces, to the 1024 real time samples produced by the IFFT operation, again in the previous sampling window and stored in the IFFT – time manipulation FPGA memory interface.

All schematic, simulation and other files for the time (and frequency) manipulation FPGA are located in the `:\Designs\Xilinx\allmanip` directory on the enclosed CD.

9.3.1 Peripheral Circuit Requirements

The processed time samples from the IFFT and addition components from the PC are stored in sequential order in their respective memory interfaces. The time manipulation block must provide addressing and read/write strobes exactly the same as for the frequency manipulation block.

9.3.2 Logic Design

9.3.2.1 Arithmetic Functions

The arithmetic functions of the time manipulation block are just a simplified version of those for the frequency manipulation block, incorporating only a single 16-bit adder. Figure 9.7 shows the circuitry for the time manipulation adder.

9.3.2.2 Timing Functions

Because there are spare CLBs in the frequency manipulation FPGA, the time manipulation circuitry can be placed within the same FPGA. Although using a Core adder each addition operation takes only one clock cycle, the timing circuitry of the frequency manipulation block can be used to address the time manipulation block's memory interfaces, thus eliminating the need for additional circuitry. Figure 9.8 shows the combination of frequency and time manipulation blocks.

9.3.3 Functionality Testing

Complete functionality testing has been undertaken on the overall time and frequency manipulation blocks. The command test file `allmanip_verification.cmd` can be re-run, or the results viewed directly from the file `allmanip.tve`.

9.4 Clock and Page Select Circuitry

9.4.1 Clock Signals

The various components of the simulator board require different clock frequencies. Three global clocks, external to the FPGAs and continuously running, are defined as follows:

- PRIMARY_CLK 70.656 MHz
- CLK_2 35.328 MHz (PRIMARY_CLK / 2)
- CLK_3 17.664 MHz (PRIMARY_CLK / 4)

These three clocks will be internally gated and divided as required within each FPGA to provide the necessary clocking for each functional block. Figure 9.9 shows the derivation of the three global clocks from an external oscillator, reproduced from the CD directory : \Designs\Xilinx\Clocks.

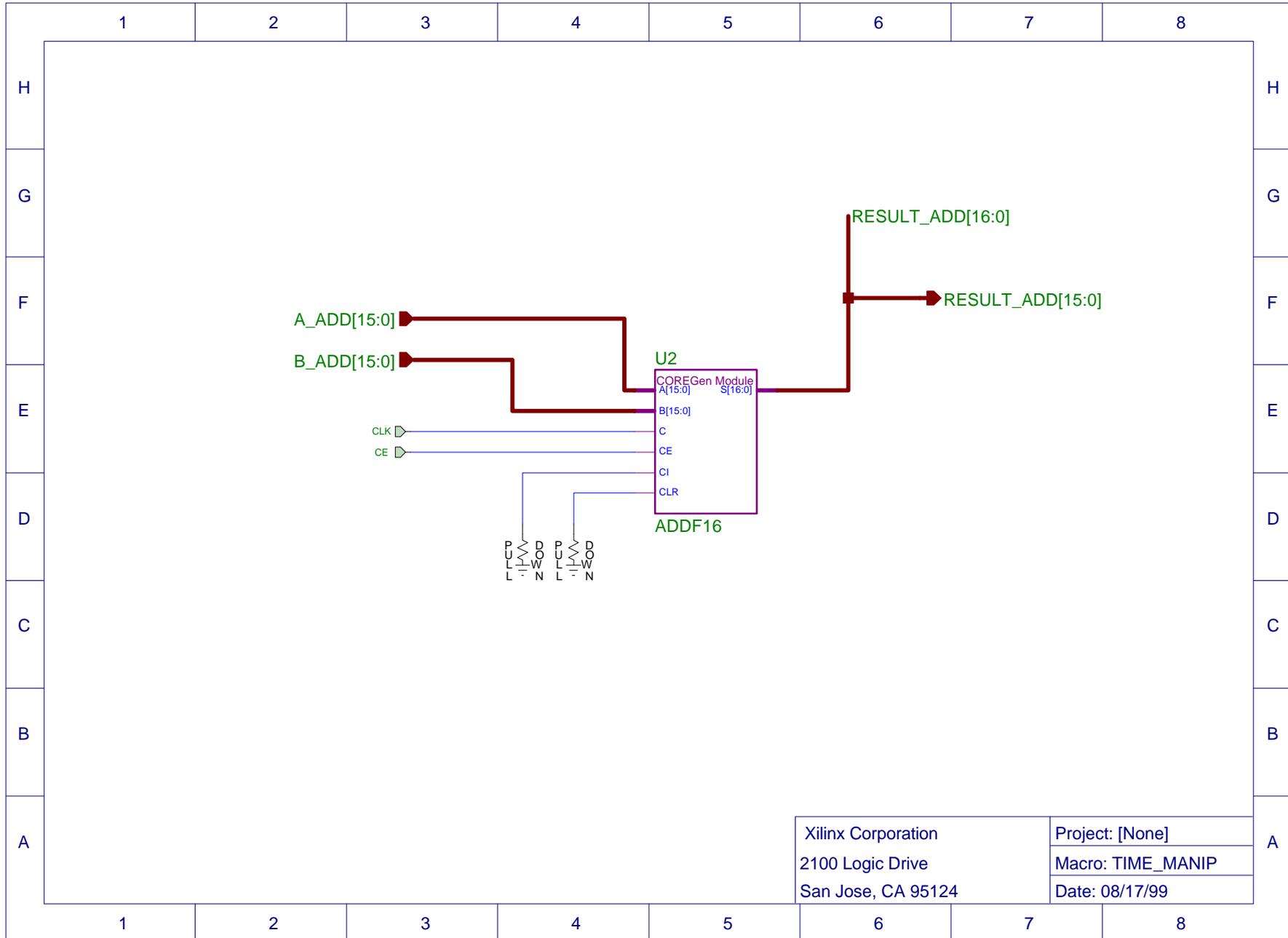


Figure 9.7 Time manipulation arithmetic functions

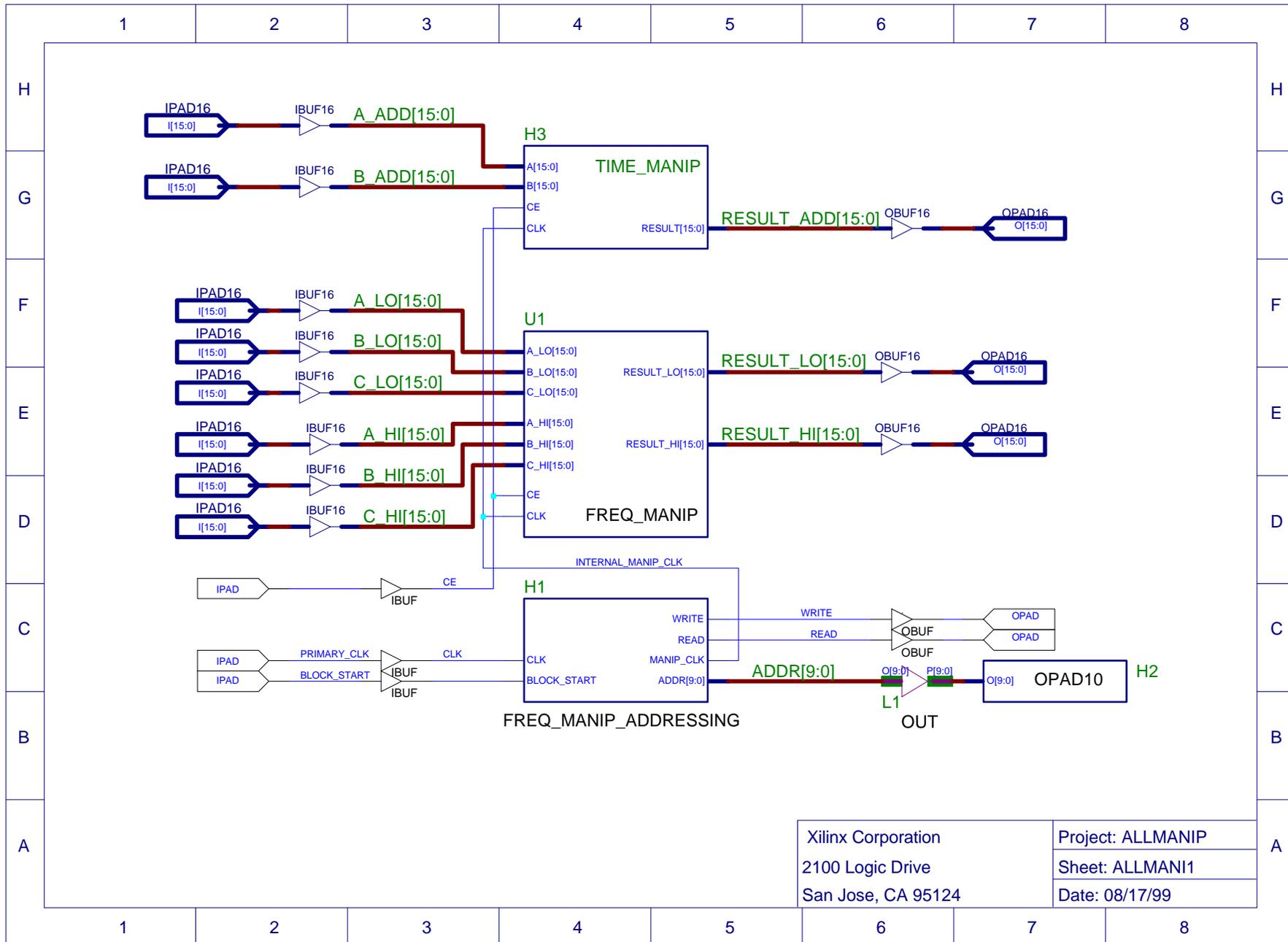


Figure 9.8 Time and frequency manipulation complete circuit

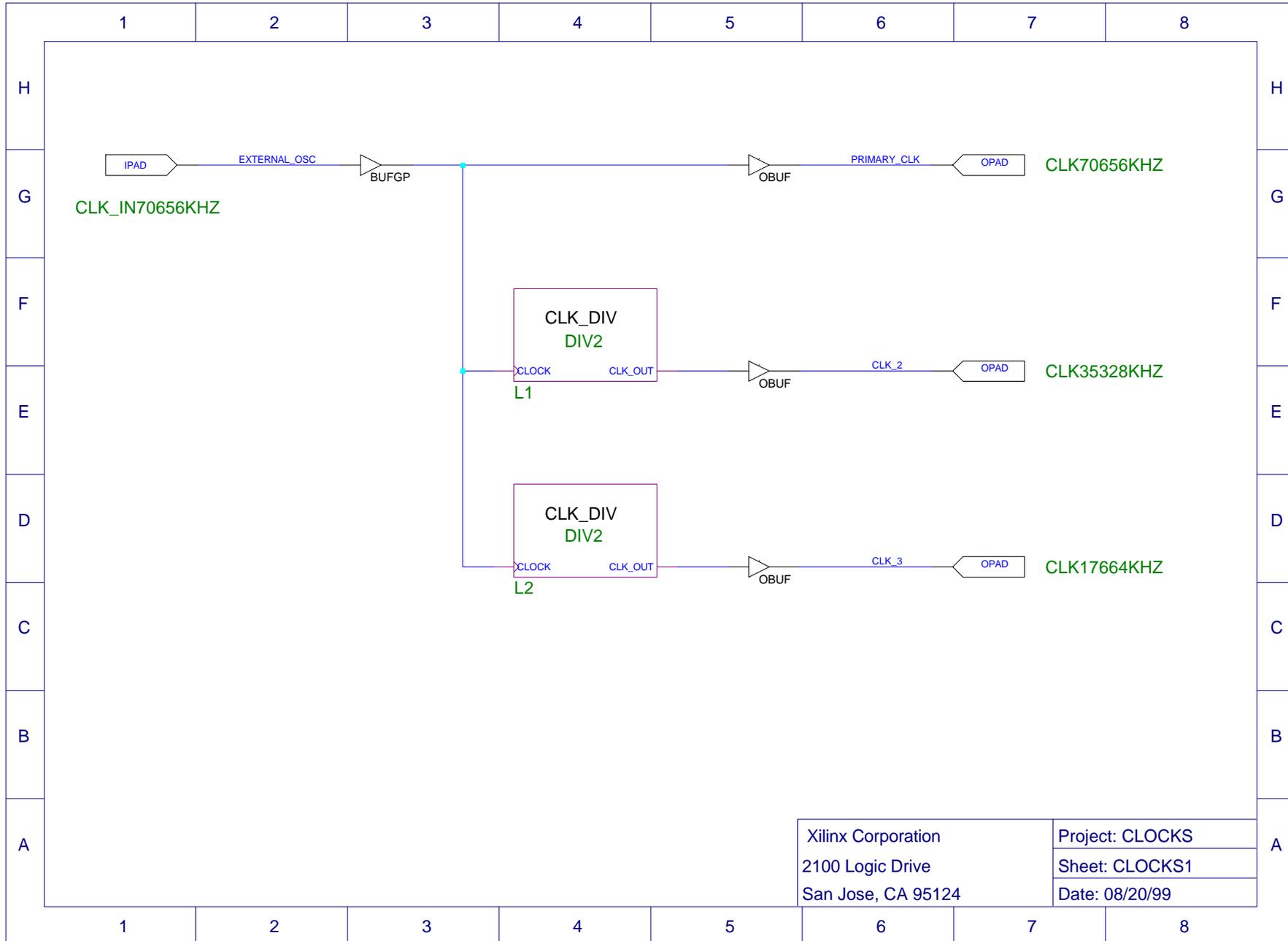


Figure 9.9 External global clock generation

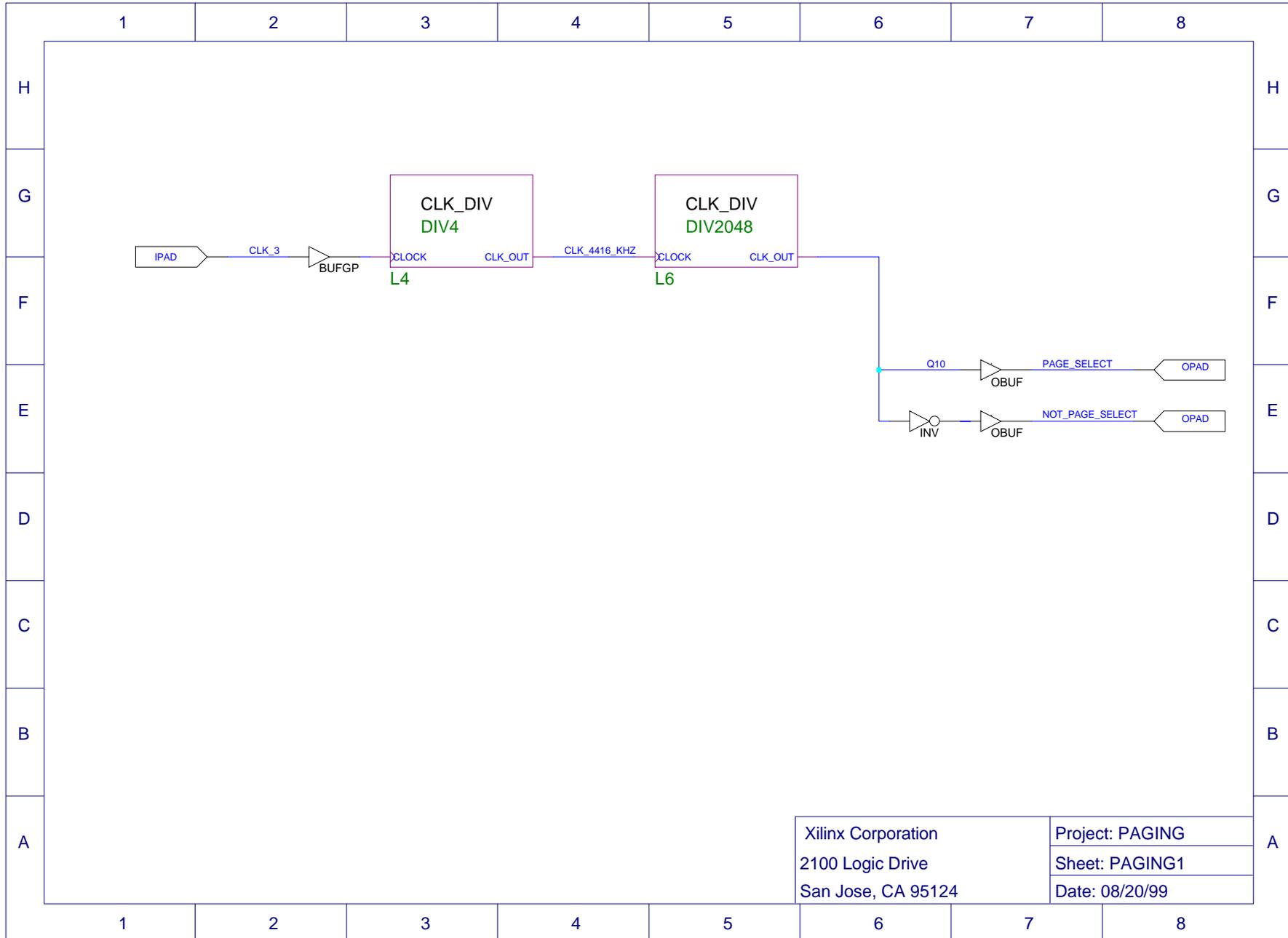


Figure 9.10 Memory interface paging circuitry

9.4.2 Memory Interface Paging

As described in chapter 8, memory interface page selection is easily achieved by dividing the 4.416 MHz ADC clock by 2048. With the use of AFEs instead of basic ADCs, the A/D conversion process clock is 17.664 not 4.416 MHz, therefore the page selection circuitry shown in figure 9.10 and stored in the directory `:\Designs\Xilinx\Paging`, initially divides the global `CLK_3` by four. The simulation command file `page_select_verification.cmd` generates the waveforms in the file `page_sel.tve`.

The appropriate `PAGE_SELECT` and `NOT_PAGE_SELECT` signals must be connected to each side of all memory interfaces, those between simulator functional blocks and PC interfaces, as shown in figures 8.1 and 8.4.

9.5 Memory Chips

Two types of memory components are needed for the simulator board: dual port RAM for memory interfaces and ordinary single port scratchpad RAM for the FFT Reference Design. The main criteria for component selection are of course access time, data word width and storage depth.

9.5.1 Interface Memory

One of the main functions of the memory interfaces is rate conversion between writing and reading from opposite sides of the RAM. Dual port RAM chips can be strobed either by a single clock on which data transfer occurs on both sides, or each side can be driven by separate clocks working asynchronously. The simulator memory must be of the latter type to allow reading and writing at different rates.

One potential problem with dual port RAM is the possibility of peripheral circuitry on opposite sides of the device addressing the same location at the same time. More expensive chips can handle this situation allowing access to the same location either during the same clock cycle through a priority mechanism or by delaying one side's access by temporarily buffering data until the next clock cycle. More basic chips have no mechanism for access conflicts. One advantage of the page structure is that locations can't be simultaneously read to and written from at the same time as the peripheral circuitry on opposite sides of the RAM are always working on separate pages of the storage space. This simplifies memory selection for the interfaces reducing device selection solely to access time and capacity considerations.

The simulator's memory interfaces can be classed into two types, those storing complex data and those for real data only. The former requires 32-bit, whereas the latter requires just 16-bit wide RAM. Regardless of the complex or real nature of the data, each interface requires a total of 2048 locations, equivalent to two pages each of 1024 locations. Recourse to the FFT Reference Design's timing

diagrams show that read and write accesses occur in single FFT clock cycles. Running on the global clock, CLK_2, the FFT interface memory must have access times better than 29 ns. The manipulation operations running at the PRIMARY_CLK require interface access times better than 14 ns

The largest manufacturer of dual port RAM is Cypress Semiconductor, producing devices of both 16 and 32-bit widths. Table 9.2 identifies two suitable devices for the real data 16-bit and complex data 32-bit interfaces.

Device	Data Width and Depth	Access Time
cy7c09389v ¹	16 bits x 16 k	7.5 – 12 ns
cy7c09579v ²	32 bits x 8 k	5 – 8 ns

Table 9.2 Suitable memory interface devices

9.5.2 FFT / IFFT Scratchpad RAM

The FFT scratchpad RAM requires 32-bit by 1024 location single port memory, operating at the FFT clocking rate. Any synchronous RAM with access times better than the FFT clocking period (29 ns at 35 MHz) will suffice. Since so many manufacturers produce suitable RAM, device selection is until board construction.

9.6 Low Level Design Summary

The FFT Reference Design with registered address and tristate data buses for scratchpad RAM use, associated input RAM address bus and control circuitry and internal clock gating can all be placed in a single 240 pin QFP XC4062XLA-07HQ240 device. Overall clocking at between 42 and 69 MHz is projected from design simulation within the Xilinx Foundation software environment. For the ADSL simulator, the global 35 MHz CLK_2 clock is used to give a complete 5206 cycle execution time of 149 μ s. Transform results appear to be incorrect, with further testing required before a definitive conclusion can be drawn. The IFFT block hasn't been designed because of the doubt concerning the Reference Design's integrity, but will be virtually identical to that for the FFT.

Both time and frequency manipulation blocks with all associated addressing and control signals for I/O memory interface data transfer have been completed. Functional verification confirms the arithmetic integrity of both operations. Both blocks fit into a single XC4013XLA-07HQ240 device. Operation at upto 125 MHz is possible, but reduced to the 70 MHz PRIMARY_CLK rate for the ADSL simulator. A complete set of 1024 frequency manipulation complex additions and multiplications require 9 632 cycles giving an execution time of 137.6 μ s. The time manipulation circuit runs from the frequency

manipulation clock to reduce logic overhead and therefore takes the same time to execute a complete set of 1024 real additions.

Paging and global external clock generation circuitry providing page selection and three free-running clock signals at 70.656, 35.328 and 17.664 MHz, can all fit within the XC4013 FPGA used for the manipulation circuits.

References

¹ Cypress Semiconductor, CY7C09569V FLEx36™ Dual-Port Static RAM, Data Sheet, February 13, 1999.

² Cypress Semiconductor, CY7C09389V Synchronous Dual-Port Static RAM, Data Sheet, November 23, 1998.