

MULTI-WRITE PORT REGISTER

DESCRIPTION

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention generally relates to multi-write port registers and, more particularly, to a multi-write port register with a local clock buffer (LCB) having a logic function controlling the register's write-enable input port and handling all the critical timing functions of the register.

Background Description

The design of multi-write port registers becomes increasingly complex and difficult as the number of simultaneous write ports required for the design increases. The design is further complicated by requirements for maskable fields or the need for set/reset/sticky-write ports implemented in the registers. A "sticky-write" of a register is a write operation that is dependent on the contents of the data being written to the register. For example, in a sticky-1 write operation, only a "1" can be written. A "0" attempted write will be repressed.

In a typical register design, Write-port Data is sent to the Write-port Bit Lines (BLs) during the CLKC cycle of the Clock signal, and the register cells to be written will have a Write Enable (WE)

signal applied to the cells during the following CLKT cycle. The register will be continually written with data coming from its Write-port BLs as long as the WE signal is asserted. Consequently, the WE signal must be deasserted at or before the next CLKC signal is asserted because the next set of Write-port Data is being sent to the BLs during this cycle. If the WE signal is not deasserted in time, there is a risk of unintentionally writing the new write data into the registers.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a multi-write port register which avoids the timing hazard resulting in over-writing good data with data from the following cycle/state.

According to the invention, there is provided a multi-write port register with a local clock buffer (LCB) controlling a pass-gate on the signal's WE input port to reduce potential timing hazards. The LCB provides flexible logic function and is controlled by a clock and word-line inputs. In addition, the LCB may also have bit-line and/or field mask inputs. The LCB handles all the timing critical functions that all the Word Lines (WLs) previously performed in conventional multi-write port registers, thus providing a substantial performance increase.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

Figure 1 is a simplified timing diagram showing the timing relationships of the clock, write port data and Write Enable (WE) signals;

Figure 2 is a schematic diagram of the write portion of a conventional multi-write port register cell;

Figure 3 is a schematic diagram showing the old multi-write port register design using the write circuit shown in Figure 2;

Figure 4 is a schematic diagram of the write portion of the multi-write port register cell according to the present invention;

Figure 5 is a logic diagram of a simple local clock buffer (LCB) used in the write control scheme of the invention;

Figure 6 is a detailed timing diagram showing the timing relationships of the system clock and the resulting LCB output;

Figure 7 is a logic diagram of an LCB which supports two conventional write ports, two reset ports and one set port;

Figure 8 is a schematic diagram of a register cell to which the LCB of Figure 7 is connected;

Figure 9 is a logic diagram of an LCB for a three-writeported register file that consists of three disjoint segments;

Figure 10 is a schematic diagram of a single

register cell of the register file to which the LCB of Figure 9 is connected;

Figure 11 is a logic diagram of an LCB that implements two reset writeports and six sticky-1 writeports;

Figure 12 is a schematic diagram of a single register cell of the register file to which the LCB of Figure 11 is connected; and

Figure 13 is a schematic diagram illustrating a w -write-port, m -entry register file (n -bit) according to a preferred embodiment of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

Referring now to the drawings, and more particularly to Figure 1, there is shown a timing diagram which illustrates the process in which an attempt is being made to write Data 1 into one of the registers of a multi-write port register. The Clock signal is divided into alternate CLKC and CLKT clock cycles. Write-port Data is propagated to the Write-port Bit Lines (BLs) during the CLKC cycle. During the following CLKT cycle, the Write-Enable (WE) signal is asserted to write data coming from the Write-port BLs to the register cells. The edges of the WE signal are supposed to be coincident with the edges of the clock signal; however, if the trailing edge of the WE signal extends into the next CLKC cycle because of clock skew and/or propagation delay, there is the risk of accidentally writing Data 2 into the register instead.

Figure 2 illustrates the write portion of a conventional multi-write port register cell having w simultaneous Write Ports. The cell itself is composed of cross-coupled inverters 21 and 22, forming a latch, followed by an output inverter 23, acting as a buffer. The register cell's Write-Enable (WE) input for each write port is shared by all the other register cells in the same row, if the register entry is not segmented into separate write fields. Figure 3 shows an example of a w -write port register file consisting of m entries by n bits. If the register entry is segmented, all the cells grouped with each write field have a common Write-Enable signal input for each of their Write Ports. Each of the cell Write-Enables are referred to herein as Write-port Word Lines (WLs). Each of the write port WLs 1 through w (associated with each entry) are connected to corresponding ones of the gates of field effect transistors (FETs) 24₁ to 24 _{n} . Write-port Bit Lines (BLs) 1 to w are connected to corresponding ones of the sources of the FETs 24₁ to 24 _{n} , and the drains of the FETs are connected in common to the input of the latch formed by cross-coupled inverters 21, 22.

For this cell, it is important that the write port WLs are deasserted before new data appears on their corresponding Write-port BLs in order to avoid writing erroneous data into the cell. This can be done by designing the WL-driver circuitry to allow for propagation delay, clock skew, and process variations so that the WLs will be deasserted at the proper time. This can be a very time-consuming iterative process that may take up additional area

and burn additional power to minimize the signal skew through the WLs.

If the register contains w simultaneous Write-ports and is divided into s disjoint Write-Fields, we will require $w \times s$ WL-Drivers. Assuming that these drivers are too large to embed within the data portion of the register, but must be placed to the side of it, $w \times s$ WL-Driver signal lines running across each register entry will be required, and this will increase the height of each cell entry in order to accommodate all of the WL-Driver signal lines within the cell.

Figure 4 shows a cell structure with a write-control scheme according to the present invention. In this figure, the same reference numerals as used in Figure 2 denote the same structure. In this circuit, a new FET, 31, has been added to the cell that now functions as the singular Write-Enable gate for the cell. A new circuit, called the Local Clock Buffer (LCB) 32, now generates the singular Write-Enable signal for FET 31. This Write-Enable signal is common among all the cells that share their write port WL signals, and are contained in the same Write Field segment. That is, the LCB 32 will drive all the cells that are situated in the same row.

It should be noted that the Write-Enable signal from the LCB 32 is the only one in this cell that is time critical; e.g., it must be deasserted before the write port BLs are updated with new data. Note that in this implementation, the write port WLs do not have any strict timing requirements, as contrasted with conventional designs where all the write port

WLs had strict timing requirements. The LCB now handles all the timing critical functions that all the WLs in the previously described design did.

Figure 5 shows the logic of a simple LCB circuit 32 that is designed for a register that has w write ports and a global write-disable signal "sgb". The combinational logic comprising NOR gates 41 and 42 generates what is referred to herein as a qualifier or select signal. NOR gate 41 receives w Write-port WLs and generates an output to one input of NOR gate 42. The second input to NOR gate 42 is a signal which inhibits writes to the register during a scan. The output of NOR gate 42 is the select signal. This signal and its inverse, generated by inverter 43, are applied to a tri-state buffer 44 which receives as its input the system clock (sysclk) signal. The output of the tri-state buffer 44 is applied to an inverter string, here represented by inverters 45 and 46, to produce a desired delay between the sysclk signal and the LCB output. In this way, the LCB signal is "tuned" for a fixed delay between the sysclk signal and the LCB output.

Referring to Figure 6, the fixed delay between the sysclk signal is shown with a timing offset between the sysclk signal and the phase one and phase two (PH1, PH2) signals. The PH1 and PH2 signals are complementary signals that rise and fall at the same time (i.e., the fixed delay from sysclk to PH1 and PH2 signals is the same for both signals). The inverse of the PH1 signal is the PH2 signal, this signal being generated by the tri-state buffer 44 if the tri-state buffer were continually enabled. The

assumption here is that data is fed by a PH1 slave latch and written during a PH2 cycle, corresponding the CLKC and CLKT cycles. The qual signal output from NOR gate 42 enables the tri-state buffer 44 to propagate the delayed PH2 clock cycle for writing data, thereby avoiding overwriting data from the following data cycle. If the qual signal is deasserted, the PH2 signal is suppressed altogether.

Figure 7 illustrates an LCB which supports two conventional writeports, two reset writeports, and one set port. An example of the register cell that receives the output of this LCB is shown in Figure 8. NOR gate 51 and inverter 52 form an OR function for the two normal writeport wordlines, while NOR gates 53 and 54 and NAND gate 55 comprise an OR network for the reset wordlines. NOR gate 56 and inverter 57 form the final OR network for the single set wordline. The reset_wl, set_wl, and LCB_Output ports in the LCB 58 are connected to the same input ports in the memory cell of Figure 8. The inputs sgb and rst1 to NOR gate 59 are global signals that can inhibit writes from occurring and can override the write requests from the set, reset, and conventional writeports.

Figure 9 illustrates an LCB for a three-writeported register file that consists of three disjoint segments: EAX, AH and AL. There are three stacks of LCBs 61, 62 and 63, each stack being associated with each of the three segment fields. There are thirty-two LCBs in each stack in this example. From a layout perspective, each LCB stack is physically located next to the register data segment (of cells) that it drives. The LCB_Output.eax port

from LCB 61 drives an EAX cell segment located within that entry set of cells, LCB_Output.ah port from LCB 62 drives the AH segment of cells, and the LCB_Output.al port from LCB 63 drives the AL segment. For this example, this schematic is replicated thirty-two times, to match up with the thirty-two entries in the register file. An example of a register file cell for one of the segments is shown in Figure 10.

In the implementation of the LCB shown in Figure 9, each of the three writeports has a segment mask signal associated with it. Writeport A is masked by `write_maskA<3:1>`, writeport B's segments are masked by `write_maskB<3:1>` and writeport C by `write_maskC<3:1>`. If `write_maskA<3>` is active, the EAX segment can potentially be written by writeport A. `write_maskA<2>` controls the AH segment, and `write_maskA<1>` controls the AL segment. Note that the `write_WLA.eax`, `write_WLB.eax`, and `WLC.eax` outputs are attached to the `write_WLA.xxx`, `write_WLB.xxx` and `write_WLC.xxx` inputs in the cell schematic of Figure 10. All of the cells that are in a data segment share these three ports. The .ah LCB outputs are attached to the AH cell segment, and the .al LCB outputs are connected to the AL cell segment.

Figure 11 shows an example of an LCB that implements two reset writeports and six sticky-1 writeports. Figure 12 shows the schematic of a register cell for this LCB. Each of the six sticky-1 writeports have a write-wordline (WL) and a bitline (BL) associated with it. If a write operation occurs, the write-WL is made active, and the write-data is

placed on the write-BL. For the sticky-1 operation, a "1" will be written into the cell if the data on the write-BL is a "1"; however, if the write-BL data is a "0", the write operation will be suppressed and the cell will retain its previous data state.

Figure 13 illustrates the invention by way of example in an w write port, m -entry register file with n bits. Each entry consists of a single segment.

If the LCBs can be imbedded within the data portion, it is possible to run only the w write port WLs plus one additional wire for each LCB output for its corresponding Write-Field. If the LCBs are located in close proximity to their attached cells, they will not require as much power to drive their outputs, or as much area. For w write fields and m entries, $w \times m$ LCBs are required in total for each register segment.

While the invention has been described in terms of several preferred embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.



CLAIMS

Having thus described my invention, what I claim as new and desire to secure by Letters Patent is as follows:

1. A multi-write port register file comprising:
a plurality of gates connected between multiple write ports and a corresponding plurality of cells of the multi-write port register file; and

a plurality of local clock buffers connected to control selected ones of said gates, the number of said local clock buffers corresponding to a number of write fields for the multi-write port register, each local clock buffer generating a write-enable signal to corresponding selected ones of said gates which write-enable signal is common among all cells contained in the same write field, said local clock buffer incorporating a fixed time delay between a system clock and an output of the local clock buffer to reduce timing hazards in writing data to the register in succeeding data cycles.

2. The multi-write port register file recited in claim 1 wherein the write ports for a cell of the register comprise a plurality of write-port bit lines and a plurality of write-port word lines, each of said write-write port bit lines being connected to a source of a corresponding field effect transistor and each of said write-port word lines being connected to a gate of a corresponding field effect transistors, the drains of said field effect transistors being



connected in common to the gate for that cell.

3. The multi-write port register file recited in claim 2 wherein the local clock buffer comprises:

combinational logic for generating a write select signal;

a tri-state buffer controlled by said select signal to pass one phase of a system clock; and

delay means connected to the output of said tri-state buffer for delaying the system clock passed by the tri-state buffer, the output of the delay means being connected to said gate.

4. The multi-write port register file recited in claim 3 wherein the combinational logic receives a plurality of write-port write lines as inputs and generates the select signal as a logical sum of signals on the write-write port write lines.

5. The multi-write port register file recited in claim 3 wherein the local clock buffer supports a plurality of conventional writeports, a plurality of reset writeports and at least one set writeport.

6. The multi-write port register file recited in claim 5 wherein said combinatorial logic receives as inputs signals from said conventional writeports, said reset writeports and said set writeport and generates the select signal as a logical sum of said signals.

7. The multi-write port register file recited in



claim 3 wherein said local clock buffer supports a multi-writeported register file consisting of multiple disjoint segment fields, there being multiple stacks of local clock buffers each associated with one of the segment fields.

8. The multi-write port register file recited in claim 7 further comprising a segment mask signal associated with each of the multiple write ports.

9. The multi-write port register file recited in claim 3 wherein said local clock buffer supports a register file with a plurality of sticky-1 writeports, each sticky-1 writeport having a write-wordline and a bitline associated with it, wherein for a sticky-1 operation, a "1" will be written into a cell of the register file if data on the bitline is a "1", but if the bitline is a "0", the write operation is suppressed and the cell will retain its previous data state.