A reconfigurable logic system using reconfigurable functional units of the type having arithmetic logic units are associated with address generating memory units. The address generating memory units having address generators that can construct addresses for memory in the address generator memory units. This frees the reconfigurable functional unit from the need to construct a sequence of addresses for the memory unit.
DPU which can generate R/W address and write data

FIGURE 2
Figure 5
Figure 7A

<table>
<thead>
<tr>
<th>Mode</th>
<th>CSM[1]</th>
<th>CSM[0]</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Address = DPU address</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Address = DPU + INCREMENT[12:0]</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Address = address</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Address = address + INCREMENT[12:0]</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Address = 0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Address = BASE[12:0]</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Address = address</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Address = address + INCREMENT[12:0]</td>
</tr>
</tbody>
</table>

Figure 7B
ADDRESS GENERATOR FOR LOCAL SYSTEM MEMORY IN RECONFIGURABLE LOGIC CHIP

DESCRIPTION OF THE RELATED ART

[0001] This application is a Continuation-in-Part of Application Ser. No. 09/343,477 filed Jun. 30, 1999, which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0002] The present invention relates to reconfigurable logic systems. Reconfigurable logic uses different configurations to have a reconfigurable chip implemented for a number of different functions.

[0003] One growing use of reconfigurable chips is in reconfigurable computing. In reconfigurable computing, an algorithm is implemented using a reconfigurable chip which has a number of reconfigurable functional units adapted to implement different functions. The reconfigurable functional units can be implemented to run a portion of the algorithm. A different portion of the algorithm can be implemented by implementing different configurations into the reconfigurable functional unit. One type of reconfigurable logic system uses reconfigurable functional units that have arithmetic logic units. These arithmetic logic units can be programmed using different configurations that implement the algorithm. In one embodiment, the data from the reconfigurable functional units can be loaded from or obtained from a memory unit on the reconfigurable logic chip. One example of a reconfigurable logic system has the addresses for the memory produced by the reconfigurable functional units.

[0004] It is desired to have an improved reconfigurable logic system.

SUMMARY OF THE INVENTION

[0005] One embodiment of the present invention is a reconfigurable logic system including a reconfigurable functional unit that is adapted to implement a number of functions and an address generating memory unit operably connected to the reconfigurable functional unit. The address generating memory unit, including a memory, has at least one address generator adapted to construct an address for the memory. In one embodiment, the address generator is able to construct a sequence of addresses for the memory unit using an incrementor. Since the address generating memory unit can produce a sequence of addresses, the reconfigurable functional unit is freed for other operations and thus the efficiency of the reconfigurable logic system is improved.

[0006] In a preferred embodiment, there are two address generating memory units: one producing the write address and one producing the read address for the memory unit. In one embodiment, the address generator is implemented including an adder to add an incremented value from a prior address. In one embodiment, a match register is used so that the generation of the address is in the first portion of the address generator and can be matched with the actual internal address provided to the memory. Also, in one embodiment, a multiplexer is used to provide multiple different input to the address generator. These inputs may include an input, an incremented value input, an address from a reconfigurable functional unit, a feedback address, and the like. The control for the multiplexer can be provided by bits in the configuration of the reconfigurable functional unit. In another embodiment of the present invention, the address generator has different modes in which, depending upon the mode, the system operates in a different fashion.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

[0007] FIG. 1 is a diagram that illustrates a reconfigurable logic system.

[0008] FIG. 2 is a diagram that illustrates a slice of a reconfigurable logic system of FIG. 1.

[0009] FIG. 3 is a diagram that illustrates an address generating memory unit of the system of the present invention.

[0010] FIG. 4 illustrates a diagram of one embodiment of an address generating memory unit of the system of the present invention.

[0011] FIG. 5 is an alternate embodiment of an address generating unit for the system of the present invention.

[0012] FIG. 6 is a diagram of a reconfigurable functional unit of one embodiment of the present invention.

[0013] FIGS. 7A and 7B are diagrams illustrating an alternate embodiment of the present invention.

[0014] FIG. 8 is a diagram of one embodiment of a reconfigurable functional unit that can be used with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0015] FIG. 1 shows a reconfigurable logic system 20. The reconfigurable logic system 20 is connected to an external memory 22. The configurations from the external memory 22, along with the data loaded by the memory controller 24 and loaded upon the system data bus 26 system address bus 28. The reconfigurable chip includes a DMA control bus interface 30, a configuration buffer 32 and the reconfigurable slices 34, 36, 38 and 40 with reconfigurable memory. Also shown is fabric setup control logic 44 and CPU 42.

[0016] FIG. 2 is a diagram of a slice that shows a number of local memories and data path units or reconfigurable functional units in a tile of each slice. Shown in FIG. 2 is a slice 50 with a tile 52, including a local memory 54, and multiple data path units 56 and 58. Also shown is a multiplier unit 59.

[0017] FIG. 3 is a diagram that illustrates a memory unit including the address unit of the system of one embodiment of the present invention. The diagram of FIG. 3 shows a single reconfigurable functional unit and local memory unit. The interconnecting pass gates 60, 62, 64 and 66 allow the connections between the local memory unit and other reconfigurable functional units (not shown). Reconfigurable functional unit 72 can produce a lead address onto line 74, and a write address onto line 76. These addresses can, depending upon the state of the tri-state buffers 78 and 80, be placed upon the write address bus 82 with a read address bus 84. These addresses can be sent to other local memory units (not shown) through pass gates 60 and 62. Local memory unit 70
includes a read address generator 86 and a write address generator 88. In one embodiment, a single address generator is shared for the reading and the writing of the memory unit.

[0018] Also associated with the local memory unit 70 is a conventional memory unit 90 which can be constructed in conventional ways of the memory. In a preferred embodiment, the conventional memory unit 90 is a multi-port memory 90. The data provided through memory unit 90 can be provided off of write data line 92 and the read data from the conventional memory unit 90 is written to the read data line 94. The data can be written to and read from by the reconfigurable functional 72 or other reconfigurable functional units not shown or connected to the read and write data lines 94 and 92.

[0019] In a preferred embodiment, an address generator, used in the local memory unit 70 can produce addresses for the conventional memory unit 90, thus freeing the reconfigurable functional unit 72 from having to continually produce the addresses. This is especially the case when a sequence of addresses is to be read from or written to the conventional memory unit 90. In that case, the address unit 86 can construct the sequence of addresses without the addresses being provided by the reconfigurable functional unit 72, or other reconfigurable functional units.

[0020] In one embodiment, the address generator includes an incrementor unit, incrementing the addresses by predetermined amounts, so that the addresses can follow a sequence to read and write in data in or out of the memory unit 90.

[0021] FIGS. 4 and 5 illustrate examples of address generators for use with the system of the present invention. FIG. 4 is a diagram of an address generator. The first portion of the address generator, includes a multiplexer 100, used to select the next address value. In the example shown in FIG. 4, the selectable inputs include a base register value is stored in a register 102, the old register value from feedback line 114, the address 106, from an address line, such as that provided by a reconfigurable functional unit, and an input from an incrementor 108 using an adder 110 and an increment register 112. Incrementor 108 allows for the address values to move through a sequence. A register 114 receives the output of the multiplexer 110.

[0022] In one embodiment, the control for the multiplexer 110 can be provided by bits associated with an output control field from a reconfigurable functional unit. This can be sent along a line, such as the lines 82 or 84, as shown in FIG. 3, or by a direct connection between the reconfigurable functional unit and a local memory unit. FIG. 5 below describes the system which has an additional mode bit controlling the mode of the address generator. Looking again at FIG. 4, the address generator also has a mask unit 116, comprising a mask register 118 and unit 120. The mask portion 116 allows the first portion 122 to construct the sequence of addresses that keeps incrementing. The mask register region 116 allows for only the lower bits to be used as the internal address to the memory. This allows the addresses from section 122 to keep incrementing while the addresses to the memory will cycle. The address on line 124 is sent to the memory portion of the system. Also showing is an ID bit register 126 that checks whether certain further bits are set, so as whether to set the enable line 128 to the memory shown.

[0023] FIG. 5 illustrates an address generator unit with the mode register 140. Generally if the mode bit is 0, the output to the next generator can an address obtained from the data path unit 142. If the mode bit is 1 then the addresses can use the base value from register 144.

[0024] Looking again at FIG. 5, if the mode bit is zero, CSM[1] is zero and CSM[0] is zero, then the input address will be the address 142 from the data path unit. Note that the output of the OR 156 will be a zero, since the CSM[1] is zero and CSM[0] is zero. This causes the multiplexer 158 to be a zero, causing the address from the reconfigurable functional unit (DPU) 142 to be provided to the zero

[0025] If the mode is a zero, CSM[1] is a zero, and CSM [0] is a 1, multiplexer 148 will provide the DPU value to the adder 150. Multiplexer 152 will provide the increment value to the adder 150. The combined DPU address and increment value are sent to the multiplexer 154. Logic element 162 provides zero to the multiplexer, a zero to the control of the multiplexer 154. Thus, the DPU and increment value is sent to the multiplexer 158. The output of the Or 156 will a 1, so that the DPU plus the increment value are provided to the register 160.

[0026] If the mode is a zero, CSM[1] is a 1 and CSM[0] is a zero, the old address value from register 160 is output from the multiplexer 148. This is added to a zero value in the adder 150. Logic 162 selects the address value from the adder 150 and provides it to the multiplexer 158. The OR unit 156 selects this address value and then sends it to the register 160. In this manner, the value from the register 160 is fed back to itself and the address stays the same.

[0027] If the mode is a zero, CSM [1] is a 1 and CSM[0] is a 1, then the address value from the register 160 is combined with the increment value and passed through the multiplexers 154 and 158 to the register 160.

[0028] If the mode is a 1, CSM[1] is a zero, and CSM[0] is a zero, logic 162 is high and the zero value is provided through multiplexers 164, 154, 158 to the register 160.

[0029] If the mode is a 1, CSM[1] is a zero, and CSM[0] is a 1, then the base value in register 144 is provided through multiplexer 164, multiplexer 154 and multiplexer 158 to the register 160.

[0030] If the mode is a 1, the CSM[1] is a 1, and CSM[0] is a zero, then the address value is fed back to the register 160.

[0031] If the mode is a 1 CSM [1] is a 1, then CSM[0] is a 1, then the fed-back address value is incremented and provided to the register 160. The breakdown of the different values is given below.

[0032] Address generator is added to the LSM;

[0033] two new CSM bits and a mode bit are needed to decode the eight available modes of the address generator;

[0034] if (Mode, CSM[1:0])=0 { address DPU address}

[0035] if (Mode, CSM[1:0])=1 { address=DPU+INCREMENT [12:0]}

[0036] if (Mode, CSM[1:0])=2 { address=address+INCREMENT [12:0]}


if (Mode, CSM[1:0]=3, [address=0])
[0038] if (Mode, CSM[1:0]=4, [address=BASE 12:0])
[0039] if (Mode, CSM[1:0]=5, [address=address])
[0040] if (Mode, CSM[1:0]=6, [address=address])
[0041] if (Mode, CSM[1:0]=7, [address=address+ INCREMENT [12:0]])
[0042] two new CSM bits;
[0043] new static configuration bits (per address generator, that means each LSM will have one for read and one for write) as described in the following;
[0044] Mode (1 bit)
[0045] Increment [12:0](13 bits)
[0046] Base [12:0](13 bits)
[0047] Mask [12:2](11 bits)
[0048] Match [3:0](4 bits)—there are Match [3:0] in CS212 shared by both read and write. For the new design, read and write will have its own Match [3:0]
[0049] Matchen [3:0]—there are Matchen [3:0] in CS212 shared by both read and write. For the new design, read and write will have its own Matchen [3:0]
[0050] 84 of configuration bits are added per row;
[0051] \[((1+13+13+11+4+4+4)>2)\text{total bits}\approx 8 \times \text{existing bits},\text{that means } 84 \times 4 \times 336 \text{ bits per tile.}

The value from the register 160 is provided to the masking portion 170. In this implementation, the bits 1 and zero of the constructed address are not masked and values go on line 172 to the memory. The mask register 174 is ANDed to the other bits with and unit 176. This ANDed value is sent along line 178 to the memory. The match register 180 and exclusive OR 182 are used to produce an enable signal so that the memory can be enabled. For example, the address may not refer to the bits 9 to 12, will indicate whether the address refers to data in that specific memory. These bits are compared to ID bits in the register for the memory of the local system memory. If the bits don’t match, then the data is not read to or written into the memory. The use of such matching bits is described in the parent application Ser. No. 09/343,477 filed Jun. 30, 1999, for LOCAL MEMORY UNIT SYSTEM WITH GLOBAL ACCESS FOR USE ON RECONFIGURABLE CHIPS, which is incorporated herein by reference. Note that the mask register 174 and the addresses can be set so as to always set the enabled bit high.

FIG. 6 is a diagram of a reconfigurable functional unit that can be used with one embodiment of the present invention.

What is claimed is:
1. A reconfigurable logic system, comprising
a reconfigurable functional unit adapted to implement a number of functions; and
an address-generating memory unit operably connected to
the reconfigurable functional unit, the address-generating memory unit including a memory and at least one
address generator adapted to construct an address for
the memory.
2. The reconfigurable logic system of claim 1 wherein the address generator is able to construct a sequence of
different addresses for the memory.
3. The method of claim 2 wherein the address generator
includes an incrementor.
4. The reconfigurable logic system of claim 3 wherein the incrementor comprises an increment register and an adder.

5. The reconfigurable logic system of claim 1 wherein the address generator includes configuration bits to control the configuration of the address generating unit.

6. The reconfigurable logic of claim 1 wherein the address generator includes a mask value to produce a masked address value.

7. The reconfigurable logic system of claim 1 wherein the addresses provided by the address generator can include an incremented value.

8. The reconfigurable logic system of claim 1 wherein the address generator such that the address provided by the address generator can be fed back the address value.

9. The reconfigurable logic system of claim 1 wherein the address is provided by the address generator can be a cycling address sequence.

10. The reconfigurable logic system of claim 1 wherein the address is provided by the address generator includes a base register value.

11. The reconfigurable logic system of claim 1 wherein the address generator includes a multiplexer to select multiple different generated addresses.

12. A reconfigurable logic system, comprising

- a reconfigurable functional unit adapted to implement a number of functions and
- an address-generating memory unit operably connected to
- the reconfigurable functional unit, the address-generating memory unit including a memory and at least one address generator adapted to construct an address for the memory, the address generator able to construct a sequence of addresses for the memory using an incrementor.

13. The reconfigurable logic system of claim 12 wherein the incrementor includes an adder.

14. The reconfigurable logic system of claim 13 wherein the incrementor includes a register to store an increment value.

15. The reconfigurable logic system of claim 12 wherein the address generator includes a masking portion.

16. The reconfigurable logic system of claim 12 wherein the address generator can provide a cycling sequence of addresses.

17. The reconfigurable logic system of claim 12 wherein the address generator includes a multiplexer to aid in selecting the address output.

18. The reconfigurable logic system of claim 12 wherein the address generator has different modes.

19. The reconfigurable logic system of claim 12 wherein the address generator has bits that allow the selection of different types of address output.

* * * * *