

XILINX BASED HARDWARE FOR PICTURE PROCESSING AND CHARACTER RECOGNITION PURPOSES

Zsolt MÁRTONFI and Zoltán SZIGETI

Department of Automation
Technical University of Budapest
H-1111 Budapest, Hungary
Fax: (+36 1) 463 2871, Phone: (+36 1) 463 3969
E-MAIL: martonfi@bme-at.aut.bme.hu
szigeti@bme-at.aut.bme.hu

Received: June 30, 1995

Abstract

The paper deals with the development of hardware that is capable realising picture processing and character recognition algorithms. The hardware was implemented as an IBM PC peripheral card and contains up to five XILINX XC3090 FPGA devices. Because of the on board reconfigurability of the XILINX devices the hardware allows to implement several separate algorithms at different times. For evaluation the performance of the hardware the Dineen and the Unger image smoothing techniques were chosen. The image smoothing techniques can be used at the pre-processing stage of the character recognition process.

Keywords: IBM PC, XILINX, programmable logic, character recognition, pre-processing, image processing.

1. Introduction

The development of picture processing hardware at the Department of Automation was initiated by Professor Gál in 1988, when an IBM PC based frame grabber card was developed (PRENCSOVSKY et al., 1989). The aim of the implemented picture processing card is to speed up parts of picture processing and character recognition software. The card has been designed to provide a cheap, efficient and flexible solution for the problem. The paper also deals with some aspects of the card development.

2. The Predecessor of the XILINX Card

We saw two ways of speeding up image processing: on one hand, to buy faster computers and, on the other, to design and implement special hardware dedicated for this purpose. At the end we decided to build an IBM PC peripheral card, which will support image processing. One of the most important requirements, beside the speed, was flexibility, so we started to de-

sign a hardware based on XILINX FPGA chips. The XILINX devices were ideal for our application because they could be reprogrammed on board.

The card was developed in 1991 at the Department of Automation as an MSc. thesis work. The card was our first attempt to implement a PC card that is capable implementing algorithms in hardware.

The card has the following parameters:

- It contains 3 pieces of XC2064 device.
- The card has no on board memory, works from and into the PC memory.
- During the memory accesses the card acts as bus master.
- The memory address generator was implemented with external counters.
- The bus arbitration was implemented in one of the XILINX chips.

Fig. 1 shows the block diagram of the card.

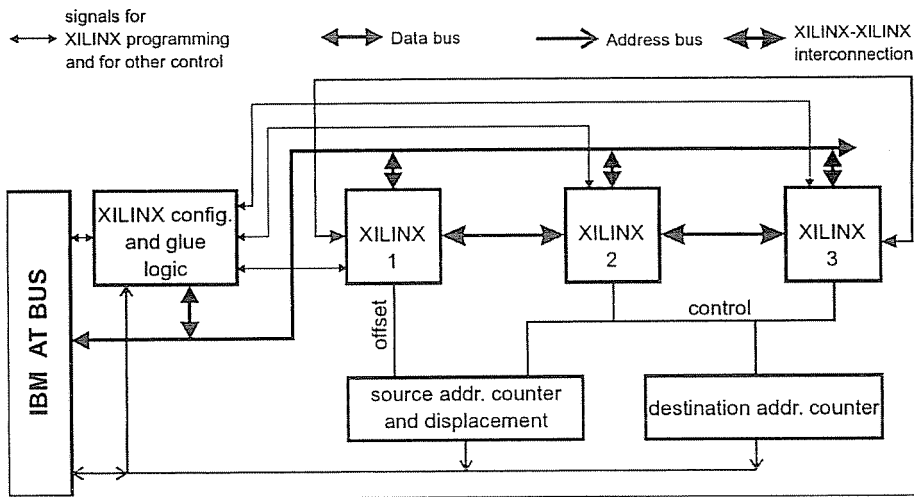


Fig. 1.

We implemented some picture processing algorithms, however, rather simple algorithms could be realised, because of the low capacity of the XC2064 devices (for example the Dineen filter (ULLMANN, 1973) fills all the three XC2064 chips). The address generator with external counters was optimal for algorithms that accessed every pixel of the image in a fixed manner, and we spared a lot of logic in the programmable devices, though it was inefficient for a couple of algorithms that required complex data transfers between the PC memory and the peripheral card. The memory was accessed directly by the hardware and we had problems with the bus mastering on

some PC clones. We could initiate only byte wide read and write operations because the data bus of the card was 8 bit wide and this reduced the efficiency of the data transfers.

Based on the conclusions we arrived using the above presented hardware we designed a more powerful and efficient peripheral card for image processing. To be able to implement more complex algorithms we used XILINX XC3090 devices for the programmable part of the new card. Using on board memory would make possible to organise the memory in banks and so to use memory interleaving for achieving higher memory bandwidth. The transfer rate between the processing logic and the on board memory can also be increased by making 32 or 64 bit wide accesses.

3. The XILINX Based Card

The new XILINX based card has the following characteristics:

- it contains up to 5 (minimum 2) XILINX XC3090 FPGA devices,
- the size of the memory can be between 512 KB and 32 MB, depending on the memory modules,
- standard SIMM memory modules,
- the control of the dynamic RAM's is realised by the DP8422A DRAM controller (DRC) manufactured by the National Semiconductor Corp. The controller provides the possibility of the dual port access,
- the card also contains the glue logic needed for programming the XILINX devices.

Fig. 2 shows the block diagram of the card.

The main modules of the card:

XILINX configuration logic:

At power up all of the XILINX devices are unconfigured, so a glue logic is needed through which the devices can be programmed. This glue logic consists of a simple I/O decoder logic and ports through which the XILINX devices can be programmed. The glue logic is used even after the configuration phase as decoder logic for I/O.ports implemented in the XILINX0 device.

XILINX0:

This device is the heart of the card, because it controls all the memory accesses from the PC side, and it addresses the memory in the picture processing mode. When data blocks are loaded into the memory of the card special interface logic can be programmed into this device to organise (on line) the data in the memory optimally for the processing stage. For exam-

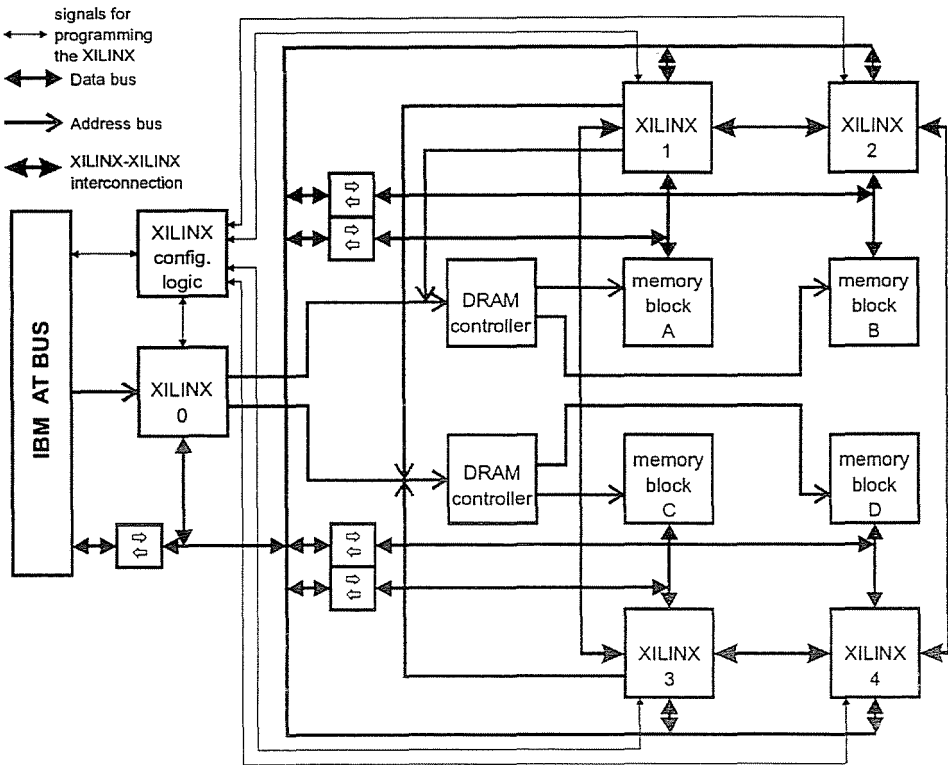


Fig. 2.

ple to spread the words in banks in such a way that memory interleaving would be possible.

XILINX1..4:

These devices are destined for the picture processing and character recognition logic. They are connected in ring and can access different memory blocks. So it is possible to implement pipeline processing, reading data at one end of the pipeline, processing it in several steps and writing into the memory at the other end. If we use separate memory blocks for source and destination data (and these memory blocks belong to different DRAM controllers) the memory read and write operations can be fully overlapped.

Memory subsystem (Dynamic RAM + DRC):

The memory subsystem stores the image to be processed. It can be accessed from the IBM PC side, and from the XILINX side, too. 16, 32, 64 bit

accesses are possible depending how many XILINX devices are present on the card. This is because every XILINX chip is related to 16 bit organised memory block. If we implement the same algorithm in all chips then 64 bits can be processed during one memory access.

4. Interfacing the Card to the IBM PC

The interface logic provides all functions needed to program the XILINX devices and the DRAM controllers, and to be able to read and write the memory from the PC side. The whole on board memory is mapped through a window in free place in the upper memory of the PC. There is the possibility to access on board memory through I/O ports, so direct memory access can be used to transfer data between the PC and the peripheral card. However, memory accesses for the data processors are also controlled by this interface logic.

Fig. 3 shows the sketch of the interface logic.

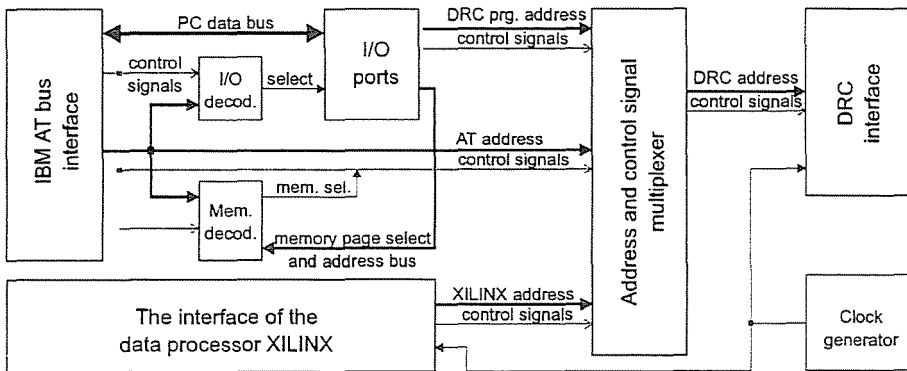


Fig. 3.

5. Implementing Image Smoothing Algorithms

Two image smoothing techniques were chosen for evaluating the image processing card for the following reasons: both techniques can use the same 3×3 window, and show the speed advantage of the hardware solution over the software one.

The Dineen's smoothing technique uses a 3×3 element window that's moved into all positions in the binary image. In each position the black elements contained in the window are counted. For each position of the

window one new pixel is generated. The destination pixel is made black only if the number of the black elements within the window exceeds a prescribed limit, Θ . The technique removes all the standalone black pixels, that are caused by noise, and high threshold levels ($5 \leq \Theta \leq 7$) produce decreasing in the limb width, weak thinning, while using the low levels ($2 \leq \Theta \leq 4$) fattens the character.

The Unger's technique applies explicit logical rules to the pattern element appearing within the 3×3 window (see *Fig. 4*).

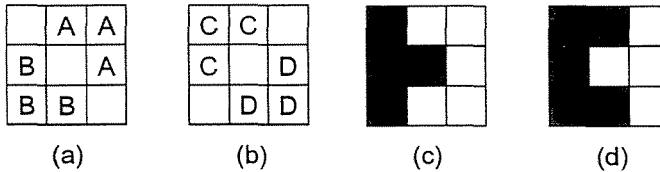


Fig. 4.

The resulting pixel becomes black if either one pixel at position A at *Fig. 4* (a) and one pixel at position B or one pixel at position C at *Fig. 4* (b) and one pixel at position D are black.

The technique removes the black pixel at the centre of *Fig. 4* (c) while changes to black the pixel in the middle of *Fig. 4* (d).

6. The Principle of the Implementation

The two smoothing techniques have the following common characteristics:

- The smoothing is performed by using a 3×3 window,
- the resulting pixel is the function of the 9 pixels of the window, so the production of the new pixel can be generated using a combinational circuit.

According to the latter sentence, the implementation of the two techniques is common, except the combinational circuit that is the core of the smoothing. The basic step of the smoothing performed by the 9 input combinational circuits is shown by the sign ① in *Fig. 5*.

The image is stored in the memory in a packed form, every pixel corresponds to one bit in the memory. This feature implies the following:

- There's a possibility to make a combinational circuit that processes 16 pixels simultaneously (the XILINX based hardware can only perform 16 bit memory access).

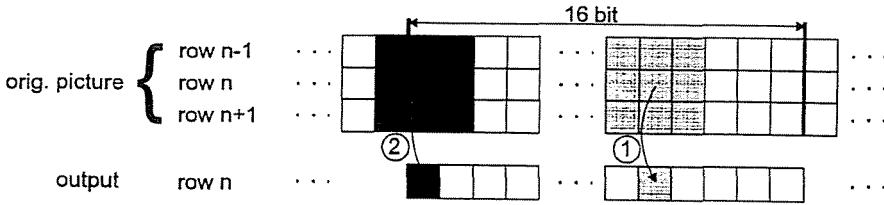


Fig. 5.

- The sign ② at the Fig. 5 shows the problem encounters at the word boundary (indicated by a thick line). For the smoothing at the word boundary, first or last bit of the neighbours must be known.

7. An Overview of the Smoothing Circuit

The Fig. 6 shows the block diagram of the smoothing circuit. The smoothing circuit was implemented on the IBM-AT compatible card.

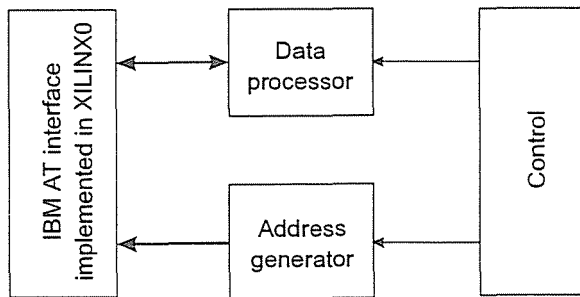


Fig. 6.

If the size of the picture is chosen to $2^n * 2^m$ the address generation can be performed by two separate counters. The address generator contains up/down counter pairs. The size of the picture is chosen to 1024*1024 pixel, this choice allows using a 10 bit counter and a 7 bit counter as address generator. The former counter acts as the row address counter, while the latter counter addresses the words within a row.

8. The Data Processor

The block diagram of the data processor for the smoothing is shown on the *Fig. 7*. The three 16+2 bit register stores the pixels of the succeeding rows. The extra two bits are necessary to perform the smoothing at the word boundary. The register stores automatically the last two bits of the previous word while storing the new value.

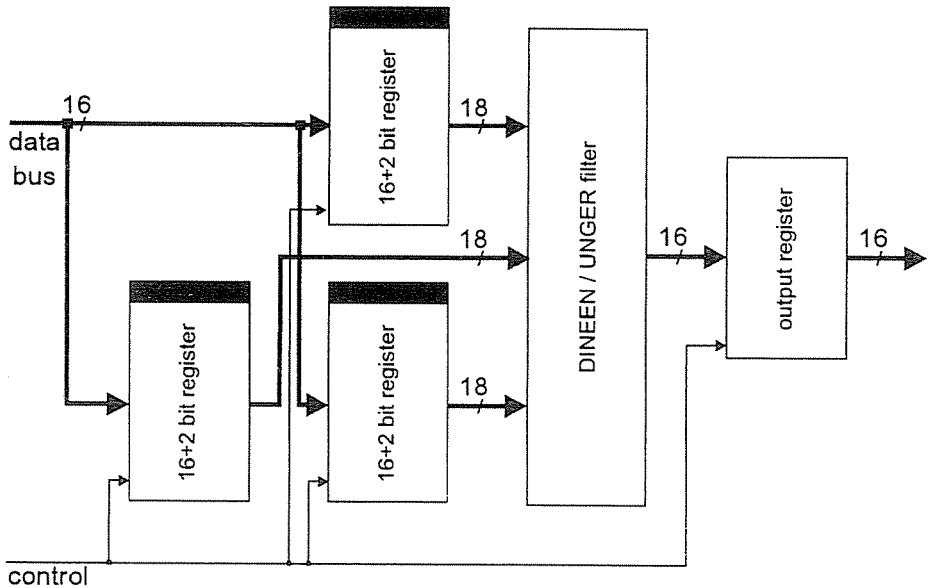


Fig. 7.

In general, both image smoothing techniques can be realised by a 9 input combinational circuit. Maximum 5 input combinational circuit can be realised with one logic block (CLB) within the XILINX LCA. Therefore the 9 input circuit of the smoothing must be decomposed to smaller, less input circuits.

The logical rules of the Unger smoothing (see *Fig. 4*) can be decomposed to four 3 input combinational circuits that check whether at the positions A, B, C, D in *Fig. 4* are at least one black pixel. Using the output of the latter circuits, a further 4 input combinational circuit can produce one pixel result of the smoothing.

The Dineen's technique uses the count of the black pixels within the windows, so it's obvious to produce the number of black pixels within a 3

pixel height column. this value can be used three times, because the 3*3 windows corresponding to succeeding pixels are overlapping.

The 16 bit wide smoothing circuit (see *Fig. 7*) can be implemented with

- Dineen's case: 34 CLBs, maximum no. of levels is 3,
- Unger's case: 48 CLBs, maximum no. of levels is 3.

9. The Timing of the Memory Access

The memory of the XILINX based card was built with SIMM memory modules. According to the data sheets of the memory modules (INTEL, 1991) the access to the dynamic RAM can be split into two phases; the active phase performs the read or write, while the passive phase called RAS precharge time. The duration of the active part (in this case) is 80 ns, while the precharge time interval is 60-75 ns. The length of the memory cycle is 150 ns. The smoothing can be finished during a precharge time, which will be 80 ns long, equal to the smoothing delay.

The timing of the memory access shown on the *Fig. 8* was done according to the following considerations:

- The last pixel of the result can be produced after reading the next three words of the input image,
- the smoothing process can be performed during a precharge time, so the memory accesses can follow each other continuously, although that precharge time must be 20 ns longer which is parallel with the smoothing,
- the refresh subsystem should be disabled, since the refresh of the memory is performed automatically by the subsequent memory accesses.

10. The Control Circuit

The control circuit provides all the control signals needed for the proper operating the data processor part, and the RAM. It's also responsible for the timings. The block diagram of the control circuit is shown on the *Fig. 9*.

The controller is microprogrammed, and it's built up from the following parts:

- **The address counter** is a loadable forward counter, it addresses the memory locations of the microprogram ROM.
- **The conditional jump control** is a multiplexer; its select lines are controlled by the microprogram ROM, and the output controls the load input of the address counter. Two inputs of the multiplexer are tied to logical 0 and 1, respectively, these are for the unconditional

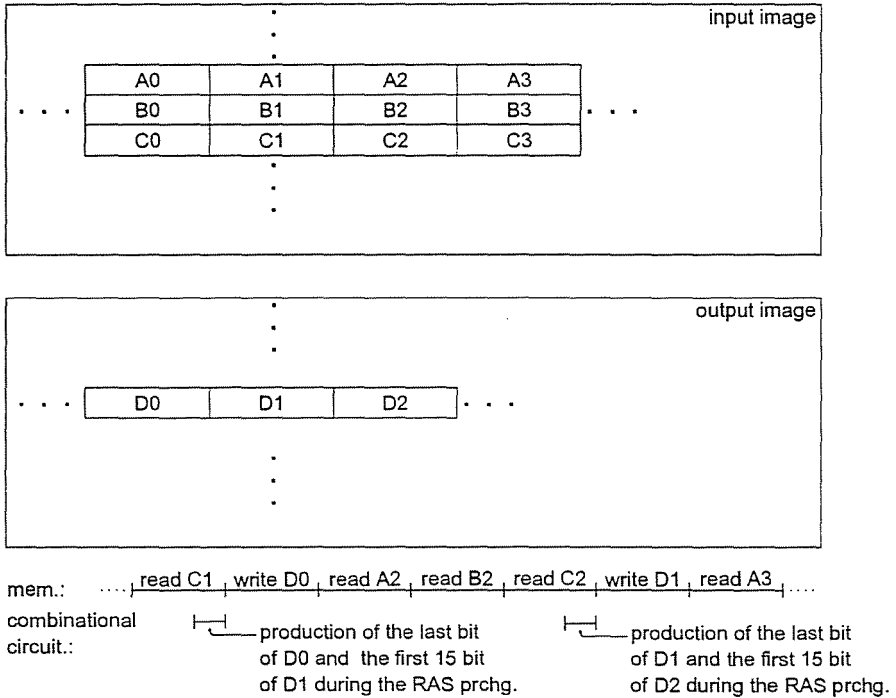


Fig. 8.

jump and sequential execution. The remainder two inputs are for handling conditions.

- The delay logic has been built from a loadable counter, which is normally deactivated. Activating the delay logic inhibits the address counter and the controller holds it's present state for a given number of clock periods. The length of the delay can be specified by the microprogram ROM. This circuit is ideal for performing fixed length delays without the waste of the memory cells in the ROM.
- The microprogram ROM has been realised the XILINX LCA. Since the combinational circuit is implemented with Look Up Tables (LUT), it can be treated as a high speed memory that's data can't be altered. With one CLB a 32*1 bit ROM can be realised, if it's not enough it can be doubled using an additional CLB and a multiplexer.

The whole microprogram ROM should be drawn on a separate sheet using hierarchical design methodology. Since the development system uses a separate netlist file (XNF) for every sheet, so the netlist file of the microprogram ROM is separated. A utility program written by the author of

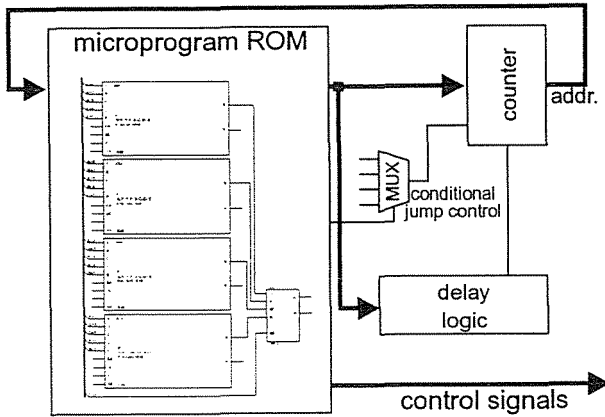


Fig. 9.

this paper is used to fill the ROM with the appropriate data. The program accepts waveform entry and alters the XNF file that describes the ROM.

11. Experimental Results

The Fig. 10 shows the experimental results. The software was running on an IBM PC with 486DX-33 processor. The program is written in assembly language, uses 8086 and 80386 (32 bit) instructions, respectively.

hardware	19 ms
program using 8086 instructions	340 ms
program using 80386 instructions	274 ms

Fig. 10.

The hardware is far more fast than any of the software solutions. The bottleneck of the hardware solution is the memory access, and so the speed of the processing depends strongly on it. Fortunately in the case of Dineen and Unger smoothing all 16 bits that are read at a time can be processed in parallel manner. The memory access can be accelerated by using cache, and can be achieved even higher speed of processing. The experiment

proves that even a comparably cheap solution can significantly speed up the compute intensive image processing or character recognition tasks.

12. Conclusion

The card that was developed at the Dept. of Automation was introduced. It's goal a comparable cheap and efficient speed up of the compute intensive algorithms like character recognition and picture processing. The architecture of the card was developed using the experience gained from a previous card.

References

1. PRENCISOVSZKY, Cs. – SZIGETI, Z. (1989): Képbaviteli modul IBM PC -hez. Végzős konferencia 1989. (in Hungarian).
2. MÁRTONFI, ZS. (1993): Hardver algoritmusok megvalósítására alkalmas eszköz készítése Diplomaterv 1993. (in. Hungarian).
3. XILINX: The Programmable Logic Data Book 1994.
4. INTEL Corp.: Memory Products 1991.
5. ULLMANN, J. R. (1973): Pattern Recognition Techniques.