

Optimization of Algorithmic Processor for Halftone Pixel Image Conversion

H. Umesh Prabhu

Department of Electrical and Electronics Engineering
St. Joseph's College of Engineering
Chennai, India

Abstract— The paper is about Modelling and synthesis of Algorithmic Processor for Halftone Pixel Image Conversion. The modelling is made based on Floyd Steinberg algorithm by which we convert M-row, N-column pixel array of resolution n bits into array having only black and white pixel. The algorithm makes the conversion by means of weighted average error of neighbouring pixels and their by converting the required pixel of n bit resolution into half tone i.e. single bit pixel. The processor is modelled first by means of Baseline design architecture and later is modelled by means of concurrent ASMD architecture and thereby we compare two architectures for an algorithm. The design is done by means of verilog HDL using Modelsim EDA tool and by means of this the design is first studied by simulation and later the processor is synthesized by means of CPLD.

Keywords— Baseline Design; Concurrent ASMD architecture; image processing; ModelSim; CPLD; Simulation; Synthesis.

I. INTRODUCTION (HEADING 1)

The photographic halftoning technique has long been used in the graphic arts and printing industries. More recently, the digital methods have started to replace the traditional photograph techniques. The digital methods offer several advantages over the traditional methods. The resultant halftone image can be displayed or printed in a computer controlled binary devices. Furthermore, it can be used conveniently for storage, transmission and further processing. Image halftoning reduces the image intensity/colour resolution for the best possible reproduction and has wide applications in the printing industry. The Halftone pixel image Conversion is needed in situation where continuous tone display can cause mismatch between image representation and display capability because of constraints of cost and system complexity and also some display devices may have lower bit resolution. Halftone also reduces bit depth and also provides Error resilient communication. In traditional mass printing media, the halftone picture is generated by the method of Screening. The screen is usually a periodic pattern made up of regular elements, each of which is graded from white to nearly black. Usually, the screen is placed in close contact with a high contrast (i.e.; high gamma) film and the process is called contact screening> the image of the photographic negative is passed through the screen and printed on the high gamma film. The image developed from the film contains only two tones namely 'black' and 'white'. The black appears as a variable size dot, with size proportional to the gray value of the original image. Previously the work is not studied by means of EDA tools. In this paper an algorithmic processor is modeled using EDA

tool and it is compared by means of two types of architecture. First model is based on Baseline design and this requires nearly equal number of processor for equal number of pixels and this leads to hardware complexity and also requires large number of clock cycles which may be time consuming. This paper compares the architecture of Baseline Design with that of another architecture called as Concurrent ASMD architecture, which is a method to reduce the complexity of hardware and reduce number of clock cycle. This architecture uses Dataflow graph by which we assign each pixel value in a row we assign a wave front index and each row of pixels are similarly assigned the wave front index and thus pixels with identical dataflow are processed together i.e. concurrently. But in order to maintain the data dependency we use a memory unit in addition to the pixel processor unit. In first section of this paper we discuss in detail about halftone images. In section II we learn what an algorithmic processor means and what high-level design is. In section III We discuss what is DFG. In section IV we discuss about the Baseline Architecture. In section V we discuss what Floyd Steinberg algorithm is all about is and finally in section VI we discuss about the concurrent ASMD architecture and their advantage over Baseline design architecture.

II. HALFTONE PIXEL IMAGE CONVERSION

Halftone images are binary images that give a gray scale rendition. For example, most printed images, including all the images printed in newspapers are halftones. The given image is oversampled (for instance, a 256*256 image may be printed on a 1024*1024 grid of black and white dots.) to coincide with the number of dots available for the halftone image. To each image sample (representing a luminance value) a random number (halftone screen is added, and the resulting signal is quantized by a 1-bit quantizer).

The output 0 or 1 then represents black or white dot. In practice the dither signal is a finite two-dimensional pseudorandom pattern that is repeated periodically to generate a halftone matrix of the same size as the image. The halftone images may exhibit Moire Pattern and the dither matrix has common or nearly common periodicities. Good halftoning algorithms are designed to minimize the Moiré effect. The gray level rendition in halftone is due to local spatial averaging performed by the eye. In general, the perceived gray level is equal to the number of black dots perceived in one resolution cell. One resolution cell corresponds to the area occupied by one pixel in the original image.

For a given sample rate of the digital image, there is a limiting number of gray levels that may be generated at a

given sample frequency. An optimum halftoning method should be able to represent the visually perceptible information of the image as closely as possible. To achieve this, different techniques of digital halftoning are proposed.

A different kind of digital halftoning is possible where each pel of the original image is mapped to say, $n \times n$ pels of binary (black and white) levels, with enough pels turned black to represent the gray level of the original image pel. The $n \times n$ pel structures or dots are decided prior and are stored in the system. In this technique, no detail is lost and the image may be displayed by utilizing the high spatial resolution capability of the binary device.

It is possible to reproduce multicolor imagery by both conventional and digital halftoning approaches. In the conventional approach, three image negatives for red, green and blue components are combined with three halftone screens and the resultant picture is printed in cyan, magenta and yellow inks. In addition, black ink is used to improve the quality of the image by the process of "under color removal". In this process, printing is done in black ink where all the three colors are present. Since printing is done by superposition, any offset in the three-screened negatives has a serious effect on the output image. Also, small relative offsets in screen angle shows undesirable Moiré pattern.

The digital approach will work in the same principle and it can be more accurate in the registration of three screened negatives. The images may be sampled and quantized at different colors by placing color filters on the original, while other operations such as color correction and halftoning is done by pel by pel basis. A single halftone screen can be used for all colors. As a result, the halftone color image printed by this technique gives a sharper image than that using the conventional technique.

III. MOIRÉ EFFECT AND FLAT FIELD RESPONSE

The practical interpolation filters results in a phenomenon called the *Moiré effect*. It appears in the form of beat patterns that arise if the image contains periodicities that are close to half the sampling frequencies. This effect occurs when the display spot size is small (compared to sampling distance) so that the reconstruction filter cut-off extends far beyond the ideal low-pass filter cut-off. Then a signal at frequency $\xi_x < \xi_{xs}/2$ will interfere with a companion signal $\xi_{xs} - \xi_x$ to create beat pattern or the Moiré effect. A special case of this situation occurs when the input image is a uniform gray field. Then, if the reconstruction filter does not have zero response at the sampling frequencies (ξ_{xs}, ξ_{ys}), scan lines will appear, and the displayed image will exhibit stripes and not a Flat Field.

II. Algorithmic Processor:

Algorithmic processors are composed of functional units, each executing in an environment of co ordinated flow. A sequential algorithm can be described by a nested loop program (NLP), which consists of a nested for loops and a loops and loop body written in a programming language (HDL) such as verilog (i.e. a cyclic behavior). NLP's are always computable, so such a starting point for a realization of a specification is attractive. Moreover, an NLP provides an unambiguous and executable specification for the machine whose behavior is to be realized.

High – Level Design:

High-level design is concerned with implementing an architecture that realizes an algorithm to accomplish in a hard-wired architecture what would be accomplished by executing a program on a general-purpose processor.

High-level design accomplishes two primary tasks:

- 1) It constructs an algorithm that realizes a behavioral specification (e.g.: create a low pass filter with given performance characteristics), and
- 2) It maps the algorithm into an architecture (i.e., a structure of FU's (functional units)) that implements the behavior in hardware. The high level design space complex because alternative algorithms may exhibit the same behavior, and because multiple architectures may implement a given algorithm, possibly with different throughput and latency.

We will focus on,

- 1) Developing an algorithm processor (i.e. a fixed architecture that implements a given algorithm)
- 2) Exploring architectural tradeoffs
- 3) Developing Virology descriptions of the architectures, and
- 4) Synthesizing the architectures.

III. DATA FLOW GRAPH (DFG)

A DFG is a directed acyclic graph, $G(V, E)$, where V is the set of nodes of the graph and E is the set of edges of the graph. Each node $v_i \in V$ represents a functional unit (FU), which operates on its inputs to produce its outputs. An FU might consist of a single operation or a more complex, ordered composition of operations, in which case the FU itself might be represented by a DFG giving a fine grained view. Each directed edge $e_{ij} \in E$ originates at a node $v_i \in V$ and terminates at node $v_j \in V$. Given an edge $e_{ij} \in E$, the data produced by node v_i is consumed by v_j .

A data dependency exists between a pair of nodes ($v_i \in V, v_j \in V$) corresponding to edge e_{ij} if the functional unit v_j uses the results of functional unit v_i and if the operation of v_j cannot begin until the operation of v_i has finished (i.e. the directed edges of the graph imply a precedence for execution). Thus, the DFG reveals the producer of data, the order in which the data will be generated, and the consumers of data. It also exposes parallelism in the dataflow, which reveals opportunities for concurrent computation and has implications for the lifetime of variables (i.e., memory). In a synchronous dataflow a node consumes its data before new data are presented. The designer's task in general, is to transform the DFG for an algorithm into a structure of hardware, typically a partitioned structure consisting of datapath unit, which, if controlled to respect the constraints implied by the DFG, will implement the algorithm. Then the control unit can be designed to coordinate the data flow of the algorithm.

Beginning with the DFG, the high level synthesis task of datapath allocation consist of transforming the DFG of an algorithm into an architecture of processors, data paths and registers from which a synthesizable *register transfer level* (RTL) model can be developed in Verilog HDL.

IV. BASE LINE DESIGN ARCHITECTURE:

A Baseline Architecture that implements a given DFG can always be formed as a set of FU's connected in a structure that is isomorphic to the DFG. This design is hardware intensive, and it serves only as a starting point because other architectures may realize the same algorithm with higher performance and less hardware. Datapath allocation binds the FU's of the DFG to a given (selected) set of datapath resources, and schedules their use of the resources.

Many different architecture can implement the same algorithm. They will be distinguished not only by their datapath resources, but also by a temporal schedule for using the resources. The high level task of resource scheduling assigns resources and a time slot to each node of a DFG. Given the parallelism of DFG, there are many schedules that could map nodes into time slots (control steps), creating several alternatives for realizing the corresponding algorithm in hardware. Scheduling must be conflict free (i.e., a resource cannot be allocated to multiple FU's in the same time slot).

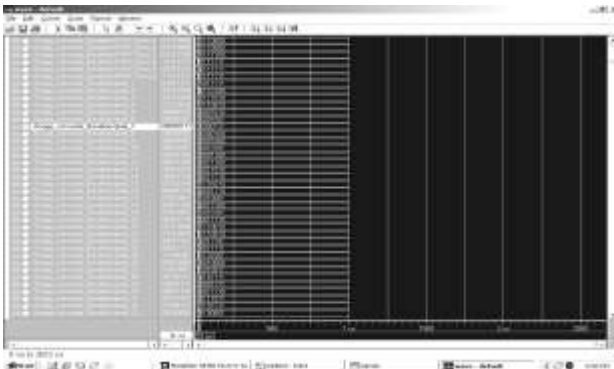


Fig.1 Simulation of Baseline Design

Approaches to reorganize the Baseline Architecture obtained from DFG:

Three general approaches are used to reorganize the baseline architecture obtained from the DFG:

1) **Recomposition:** This segments the FU into a sequence of functions that execute one after the other to implement the algorithm. The sequences of execution may further be distributed over space (hardware units) and time. In the former case, the nodes of the DFG are mapped isomorphically to the FU's in the latter case a single FU executes over as many cycles as required to complete the operations represented by the DFG. This approach saves hardware by replicating the activity of a single FU over multiple time steps, rather using multiple processors that execute concurrently in a single step.

2) **Pipelining:** Pipelining inserts register into a datapath to shorten computational paths and thereby increase the throughput of a system, incurring a penalty in latency and the number of registers in contrast to recomposition.

3) **Replication:** Replication uses multiple, identical, concurrent executing processors to improve performance but at the expense of hardware.

V. FLOYD- STEINBERG ALGORITHM

The Floyd- Steinberg Algorithm converts an image consisting of N-row*M-column array of pixels, each having

only black and white pixels, while incorporating a subjective measure of quality of the image.

The algorithm distributes to a selected subset of each pixel's neighbors the roundoff error induced by converting the pixel from a resolution of n bits to a resolution of 1-bit. The distribution of error to a given pixel is based on a weighted average of the errors at the sites of its selected neighbors. A pixel receives a distribution of error from its neighbors. The errors received from the neighboring pixels are used to calculate the halftone pixel value at cell (i, j).

Where 'i' is the column index and 'j' is the row index of the N*M array having its origin at the upper left corner of the array (a common reference for images). These relationships hold for each pixel in the array leading to the DFG.

The data dependencies of the array reveal that the pixels can be converted in a sequential manner, from left to right, from top to bottom, beginning at the top left corner and proceeding to the bottom right corner.

The FU (nodes) of the DFG execute the pixel conversion, according to the following pseudo code description of the sequence of calculations that will ultimately be described by a cyclic behavior in Verilog. At each pixel location, (i, j) a weighted average of the (previously calculated) error e, at the selected neighbors is formed according to,

$$E_{av} = (w1 * e [i-1, j] + w2 * e [i-1, j-1] + w3 * e [i, j-1] + w4 * e [i+1, j-1]) / w_T \quad (1)$$

Where,

E_{av} – Average Error Value

$w1, \dots, w4$ are subjective nonnegative weights and $w_T = w1 + w2 + w3 + w4$.

This weighted average is used to calculate a corrected pixel value (CPV) for each pixel value PV[i, j]

$$CPV = PV [i, j] + E_{av} \quad (2)$$

And then we round CPV to 0 or 1, according to a threshold, CPV_thresh. So,

$$CPV_{round} = 0 \text{ if } CPV < CPV_{thresh} \quad (3)$$

And

$$CPV_{round} = CPV_{max}, \text{ otherwise} \quad (4)$$

Where $CPV_{max} = 255$ for an image with 8-bit resolution, and $CPV_{thresh} = 128$. Next we form a halftone pixel value (HTPV) and save the error Err [i, j]

$$HTPV = 0 \text{ if } CPV_{round} = 0, \text{ otherwise } HTPV = 1; \quad (5)$$

$$Err [i, j] = CPV - HTPV \quad (6)$$

VI. CONCURRENT ASMD-BASED ARCHITECTURE:

The Baseline design model is hardware-intensive and structural, consisting of a systolic array of 48 identical processors (for 8*6 array of pixels) hard-wired for the dataflow of the DFG. The longest path through the array will limit the cycle time of the host processor. The implementation is synthesizable as combinational logic and requires no controller.

An alternative hardware implementation of the image converter algorithm is by partitioning the image converter into an algorithmic state machine with a datapath (i.e., ASMD). The data dependencies that are evident in the DFG reveal parallelism that can be exploited to reduce the number of clock cycles and the number of processors required to update the array, at the additional expense of requiring memory. In this architecture we define a computational wave front (i.e., a locus of DFG nodes that can execute concurrently in a given time step). Each node is annotated with a wave front index denoting the time step in which it may execute. The array of nodes can still be processed sequentially, in the order of ascending indexes, but nodes with an identical wave front index can execute concurrently (i.e., in the same time step).

The wave front indexes of the DFG partition the temporal domain and identify clock boundaries. Within a clock boundary, the graph reveals the resources required by a synchronous machine that would implement the processor and exploit the parallelism. The machine would update the entire image in only 18 time steps (for 8*6 array), while using fewer FU resources and requiring design of a more complex controller than the baseline design.

The architecture of the machine's datapath will be designed to exploit the concurrency that is evident in the DFG. Then a controller will be designed for the datapath. This paper pursue by manual methods what a behavioral synthesis tool should accomplish and then compare the result to the baseline design in more detail.

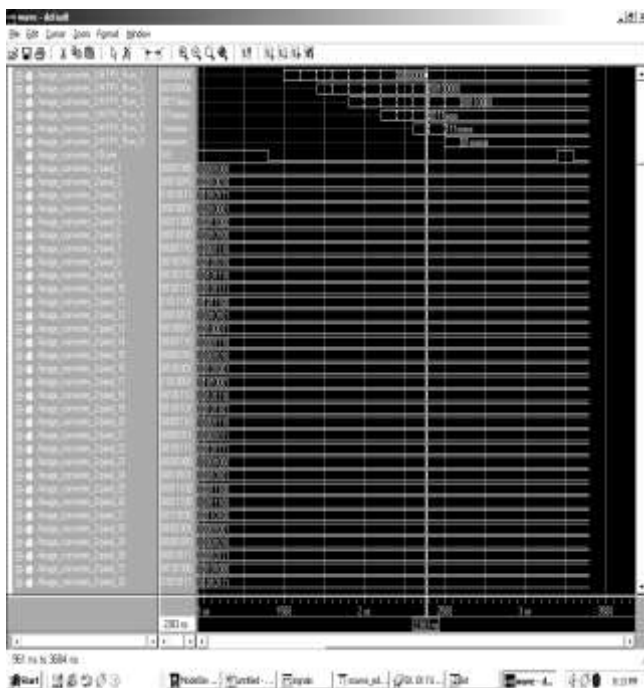


Fig.4. Simulation of Concurrent ASMD Architecture

A Comparison of alternative halftone pixel image converters
TABLE I

Tradeoffs: 8*6 Pixel Halftone Image Converter			
Version	FU Utilization	Memory Utilization	Execution Time
Baseline Design ¹	48	None (Combinational)	T _{Baseline}
Concurrent ASMD	4	6+2*48*8 bytes	18*T _{FU} (12*T _{FU})**

**Streaming Images
1 NLP-based Structure of FUs

TABLE II
Reservation Table for Mapping DFG nodes to Time slots and Processors

		TIME SLOTS																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
PROCESSORS	P1																		
	P2																		
	P3																		
	P4																		

VI. CONCLUSION

The weighted averaging method used in Floyd-Steinberg algorithms is capable of reproducing a complete tone scale, also half toning renders image sequences onto display devices that have limited intensity resolution and colour palettes. It provides an alternative for image representation, rendering, storage and transmissions in places where continuous tone.

A. Abbreviations and Acronyms

- HDL – Hardware Descriptive Language
- ASMD – Algorithmic State Machine and Datapath
- CPLD – Complex Programmable Logic Devices
- EDA – Electronic Design Automation

REFERENCES

- [1] B.E. Bayer, An optimum method for two level rendition of continuous tone pictures Modelling design and control of flexible manipulator arms. A tutorial review, *Proc. IEEE Int Conf.Communication, on Decision and Control*, vol 1, 1973 pp 11-15.
- [2] J. Sullivan, L. Ray and R.Miller, "Design of minimum visual modulation halftone patterns." *IEEE Trans Syst.,Man, Cybern, on Decision and Control*, vol 21, no.1.pp33-38, Jan./Feb 1991. 1973 pp 11-15.San Francisco, CA, 1990, 500-506.
- [3] Zhaohui Sun, "Video Halftoning" *IEEE Trans, on Image Processing*, vol.15,.No.3 March 2006 pp 678-685..
- [4] R.Floyd and L.Steinberg "An adaptive algorithm for spatial gray scale". *Proc. Soc.Inf.Display*, vol 17, No. 2, pp 75-77, Mar 1976.
- [5] B.B. Chaudhuri and D. Datta Majumder, *Two-tone image processing and recognition* (Wiley Eastern Limited, New Delhi : 1993).
- [6] Michael D.Clietti *Advanced digital design with the verilog HDL* (Pearson Education (Singapore) Pte Ltd.; India: Prentice Hall 2003).
- [7] Sybil Ihrig and Emil Ihrig, *Preparing digital images for print* (McGraw-Hill; California U.S.A: 1996
- [8] Gonzales and Woods *Digital image processing* (McGraw Hill; USA: 2002)