SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

THINNING DIGITAL PATTERNS USING THE IMSA110

1. INTRODUCTION

1.1 Thinning

Thinning is a very important preprocessing stage of pattern recognition. It is a technique which extracts the basic shapes from images. These shapes are known as skeletons. It attempts to remove all redundant points while maintaining the basic structure and connectivity of the original patterns.

In [1] an algorithm is proposed and modifications to it are described in [2]. The final form of the algorithm has the advantage of being both fast, and suitable for parallel operation.

1.2 The IMSA110

The IMSA110 [3] is a single-chip reconfigurable and cascadable subsystem suitable for many high speed image and signal processing applications.

The IMSA110 consists of a configurable array of multiply-accumulators, three programmable 1120 stage shift registers, a versatile post-processing unit and a microprocessor interface for configuration and control purposes.

Figure 1 shows the main processing core of the device. It consists of 21 multiply accumulate stages arranged in three banks of seven. These may be configured as either a 21 stage pipeline or a 3x7 two-dimensional window. The output from the MACs is fed into the backend processing unit. It is this section which allows various data transformations to take place adding greatly to the flexibility of the overall device. The maximum data rate which may be applied to the inputs is 20MHz.

Figure 2 shows a functional block diagram of the backend post processing unit. Complete details may be found in [3].
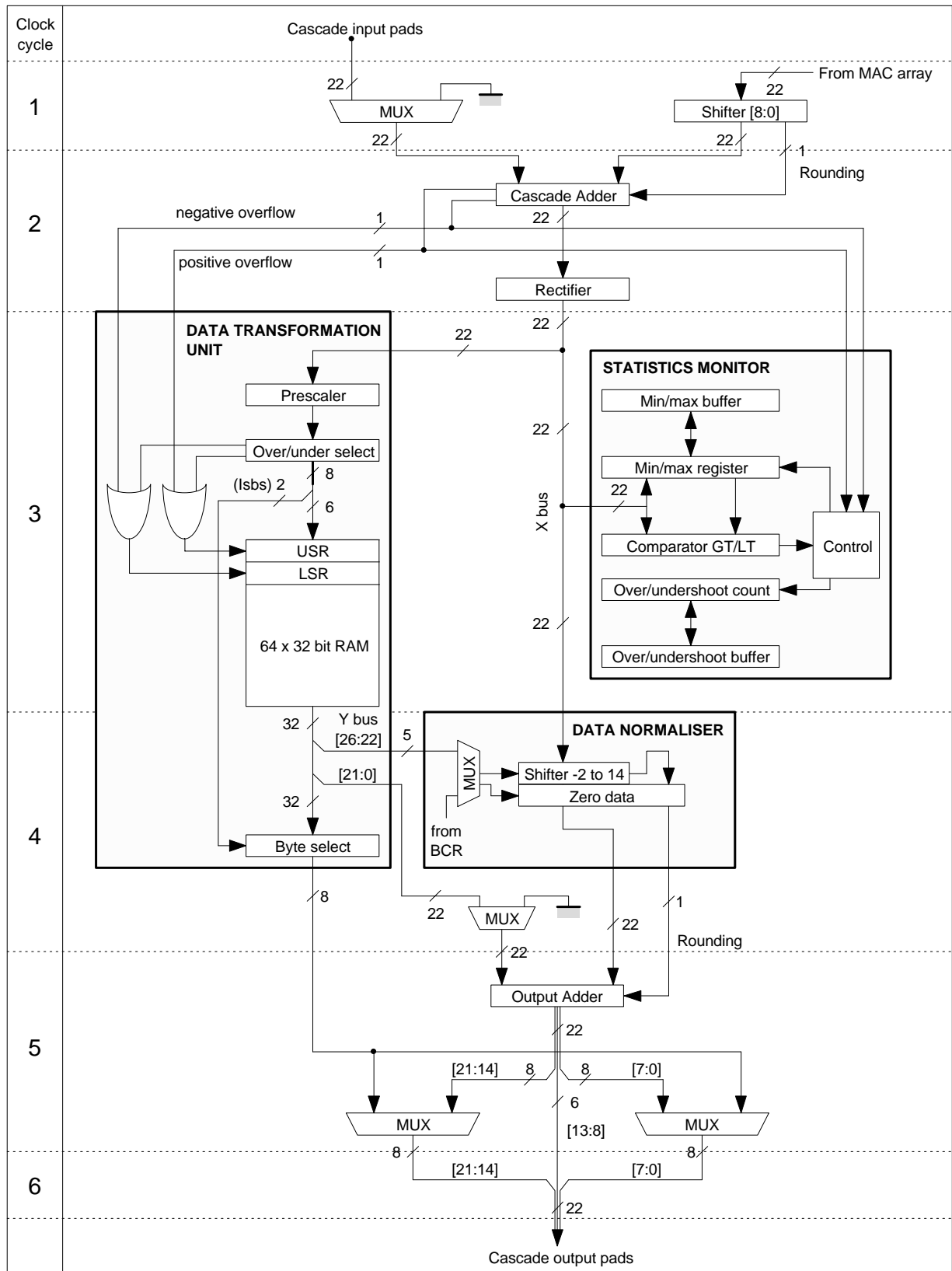
Figure 1 : IMSA110 Users Model
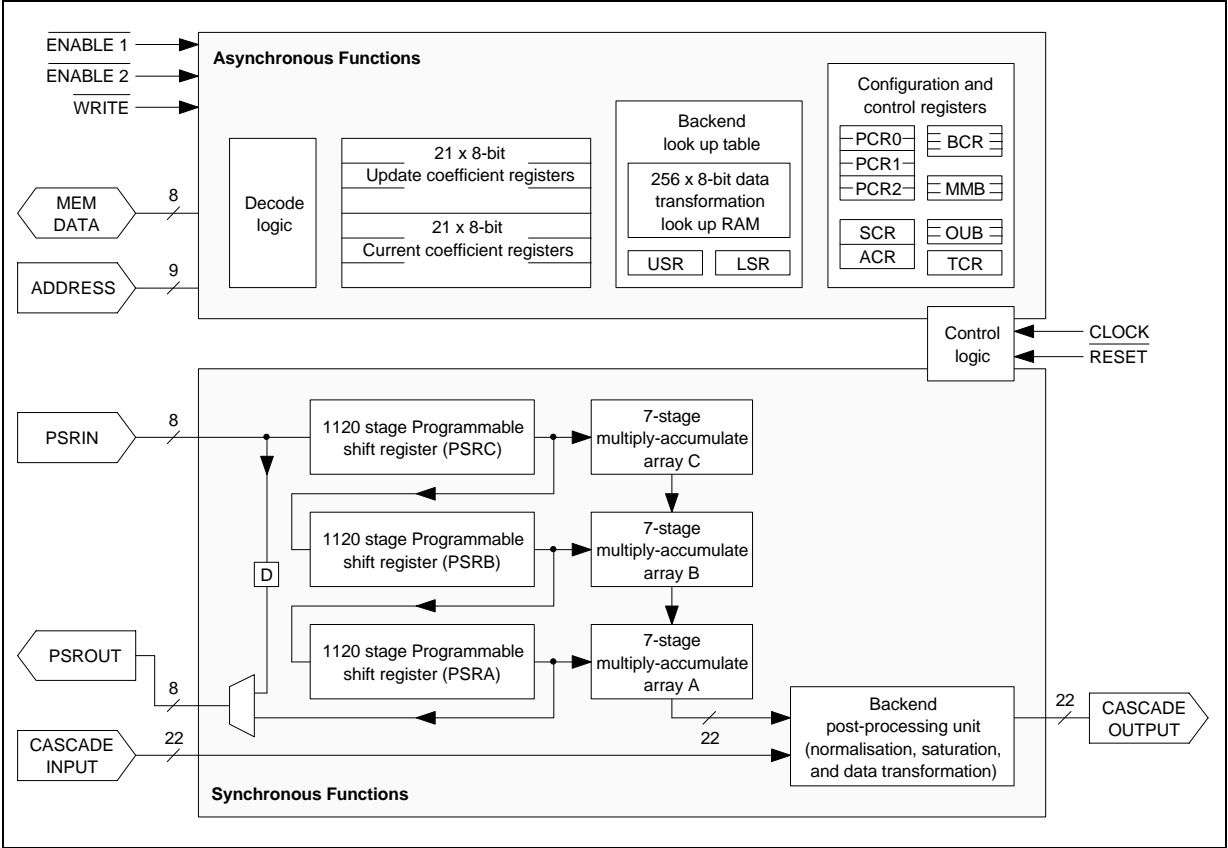
SGS-THOMSON
MICROELECTRONICS

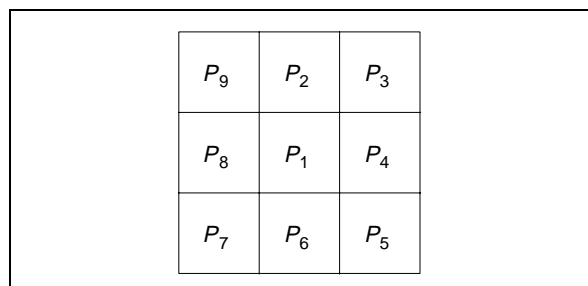**Figure 2 :** Detailed Block Diagram of the Back-end Post Processing Unit

SGS-THOMSON
MICROELECTRONICS

2. THE ALGORITHM

Consider an image Im in which every pixel Im(i,j) is either 0 or 1. It is normal for 0 to represent the background and 1 to represent the foreground patterns. It is assumed that each pixel Im(i,j) has eight closest neighbours as shown in Figure 3.

Figure 3 : A Pixel P_1 and Its 8 Closest Neighbours

The output of the algorithm for any given pixel only depends on the value of that pixel and its eight nearest neighbours. This allows parallel processing to be applied with the possibility of all the picture elements being processed simultaneously.

The nature of the algorithm is iterative, each iteration takes the image closer to the fully thinned result. When an iteration is performed which doesn't cause any change to the image then nothing further can be gained by applying further iterations. Each iteration is divided into two subiterations which erode the pattern or patterns to be thinned on opposite edges.

In the first subiteration the pixel P_1 is deleted if all of the following criteria are satisfied:

- $3 \leq B(P_1) \leq 6$
- $A(P_1) = 1$
- $P_2.P_4.P_6 = 0$
- $P_4.P_6.P_8 = 0$

Where $A(P_1)$ is the number of 0 to 1 transitions around the closed path $P_2..P_9$ and $B(P_1)$ is the number of non zero neighbours of P_1.

The second subiteration is identical to the first except that the last two criteria are changed to:

- $P_2.P_4.P_8 = 0$
- $P_2.P_6.P_8 = 0$

It should be noted that this algorithm is not perfect. Some digital patterns will totally disappear. In fact any pattern that can be reduced to a 2 by 2 square will disappear entirely. A solution to this problem will be presented in section 4.

3 DISCRETE IMPLEMENTATION

A number of methods of implementing thinning are available. These typically involve the use of arrays of processors such as the ICL DZP (See [5]). Such methods are expensive and physically large but do allow more complicated algorithms than the one presented in the previous section to be implemented.

A simple binary thinning unit which may be built in hardware is shown below. Such a unit is capable of implementing one subeteration of the algorithm described in section 2. It works by arranging for each of the address inputs of the PROM to correspond to one of the pixels in a three pixel square region. By programming the PROM with the appropriate data, which may be calculated from the specified criteria, the output of the PROM gives a new image in which the objects should have been erolded. This process is repeated until no further erosion takes place (note that alternate iterations must use alternate sets of criteria sets of criteria to obtain an unbiased operation).

The performance of such a unit is considerable and it should be fairly trival to process ten million pixels per second. In addition the unit is cascadable which can considerably increase the performance of a system. It does have a number of disadvantages however :

- it is not a single chip solution, unless use is made of semi/full custom chip design.
- it is inflexible. Replacing the PROM with SRAM would improve matters but even so the range of functions such a unit can perform is very limited.

Figure 4 : An Alternative Hardware Implementation
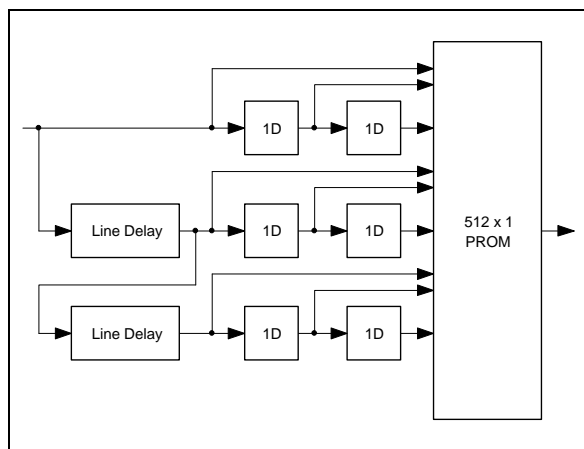
SGS-THOMSON
MICROELECTRONICS

4. A110 IMPLEMENTATION

4.1. The basic implementation

Consider the 2-D filter kernel shown below. When a binary image composed solely of O's and 1's is applied to this kernel the output consists of numbers in the range of 0 to 511. Each output uniquely identifies the pattern of O's and 1's which caused it. By feedit this output into a look up table which has 512 entries it is then possible to generate the output value for P1. The look up table must be programmed with the appropriate pattern of O's and 1's as defined by the criteria for one of the subiterations.

Figure 5 : A Kernel For Binary Thinning

Unfortunately such a kernel cannot be implemented on one IMSA110. There are two reasons for this:
- The LUT only contains 256 entries plus the upper and lower saturation registers.
- Only coefficients between -128 and 255 mays be programmed into the MAC.

The first problem may be overcome by inverting the input image and programming the upper saturation register to 1 so that now any pattern with P1 deleted will give an output which is at least 256. This will cause an overselect to occur at the prescaler. The effect of this is for the LUT output to be taken from the USR (Upper saturation register). Thus the output of the LUT will be 1 which indicates a deleted pixel in the inverted image convention.

The second problem coulb be overcome by using two IMSA110s. This is achieved by superimposing the two kernels below (See [4] for details about cascading). It would be desirable however to implement each subiteration in a single device.

Figure 6 : Twin Kernels for Binary Thinning

By inspection of the criteria it may be seen that if the coefficients are changed to the kernel shown below then it is still possible to produce a valid look up table. This occurs because although there are two different patterns which can give a MAC output of 127 the required LUT output for each is the same. This method allows a single IMSA110 to fully implement one subiteration of the thinning algorithm. Section 7 shows the data to be programmed into the IMSA110 to implement this algorithm.

Figure 7 : A Kernel For Single IMS A110 Binary Thinning

### 4.2. A solution to the disappearing pattern problem

As mentioned in section 2 any object which thins down to a 2 by 2 pixel square will disappear entirely. This problem may be overcome by slightly modifying the data in the lookup tables.

The first subiteration attacks objects from both below and to the left. In order to stop 2 by 2 pixel regions disappearing it is necessary to stop the elimination of the central pixel in the image segment shown below. Note that white indicates a pixel which is set. o do this the date at location 1F0 in the first subiteration must be set to 0.

SGS-THOMSON
MICROELECTRONICS

Figure 8 : The central pixel in this image
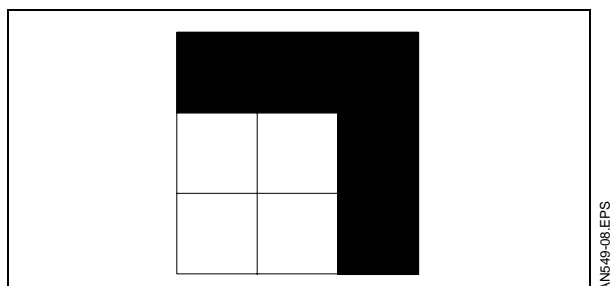
An identical procedure may be applied for the second subiteration except that in this instance the date at location 11F must be set to 0.

4.3. Monitoring the progress of the operation with the statistics monitor

The completion of the thinning operation may be detected by using the statistics monitor in the backend. The procedure for doing this with a single IMS A110 is as follows:

- Set the max register MMR to 254, and configure the IMSA110 statistics monitor as an overshoot counter.
- Zero the over shoot counter (OUC).
- Perform the first subiteration.
- Record the contents of the OUC register which now indicates the number of pixels already deleted at the start of the iteration.
- Perform the second subiteration.
- Repeat from tep 2 for the next iteration, if the same value is obtained in the OUC register twice in succession then the thinning operation is complete since no further pixels have been deleted.

The actual change in the overshoot count for each iteration may be used as an indication of the amount of progress being made.

5 PERFORMANCE ASSESSMENT

To perform a binary thinning operation on a typical 512 pixel square image using a single IMSA110 would take just over 13 ms for each subiteration at 20 MHz (this neglects the time spent reconfiguring the look up table for the next subiteration). Thus if 12 subiterations were required to fully thin an image then this would take about 156 ms. This performance level is formidable byt mey be easily increased by cascading a number of devices together (See [4]). For example if two devices were cascaded so that the first and second devices performed the first and second subiterations respectively then each complete iteration would still take just over 13ms. So to apply 12 subiterations with this configuration would require only 78ms. This principle may be extended to cascades containing any number of devices (even numbers are preferred since no lookup table reconfiguration is required). If 12 devices were cascaded then the complete thinning operation would take only 13ms. In addition the screen may be sliced up with separate portions being sent to different IMSA110s or cascades of IMSA110s for even greater performance.

The IMSA110 offers other advantages when built into a system since it is capable of performing all the initial image processing commonly associated with image recognition.

- Preprocessing filtering.
- Edge detection.
- Thresholding.
- Thinning.
- Pattern matching.

6 CONCLUSION

It has been shown how the IMSA110 may be used to provide a very high performance and expandable thinning engine. This when coupled to the other abilities of the IMS A10 make the device ideal as a front end processor in many image processing or recognition systems.

7 IMPLEMENTATION DATA

7.1. First subiteration

SCR	090	5C
ACR	092	00
CR0a	000	40	7F	01
CR0b	010	20	FF	02
CR0c	020	10	08	04
PCRa	080	Line length +7
PCRb	082	Line length +7
PCRc	084	A suitable value to deskew the output image
BCR0	0A0	01
BCR1	0A1	00
BCR2	0A2	40
BCR3	0A3	80
USR	0F8	00	00	00	01

LUT	100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

ACR	092	02

7.2. Second subiteration

SCR	090	5C
ACR	092	00
CR0a	000	40	7F	01
CR0b	010	20	FF	02
CR0c	020	10	08	04
PCRa	080	Line length +7
PCRb	082	Line length +7
PCRc	084	A suitable value to deskew the output image
BCR0	0A0	01
BCR1	0A1	00
BCR2	0A2	40
BCR3	0A3	80
USR	0F8	00	00	00	01

LUT	100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

ACR	092	02

8. REFERENCES

1 T.Y. Zhang, C.Y. Suen _ A fast parallel algorithm for thinning digital patterns. Comm. ACM 27, 3.

2 H.E.Ly, P.S.P. Wang _ A comment on "A fast parallel algorithm for thinning digital patterns". Comm. ACM 29, 3.

3 IMSA110 image and signal processing sub-system datasheet.

4 Cascading IMS A110s application note.

5 C.M. Holt, A. Stewart, M. Clint, R.H. Perrott. An Improved Parallel Thinning Algorithm. Comm. ACM 30, 2.

SGS-THOMSON
MICROELECTRONICS