

A MULTIPLIERLESS RECONFIGURABLE RESIZER FOR MULTI-WINDOW IMAGE DISPLAY

Ching-Mei Huang, Tian-Sheuan Chang and Chein-Wei Jen
 Department of Electronics Engineering
 National Chiao-Tung University, Hsinchu, Taiwan, R.O.C.

ABSTRACT

This paper presents a real-time resizing IC that can dynamically reconfigure the multiplierless polyphase CIC (cascaded-integer-comb) filter modules to meet even non-integer resizing ratio. The hardware cost is greatly reduced by using overlap-save based block input and concurrent register reset scheme. The simulated results show that this chip can process four 320x200 30 frames/sec at 55 MHz clock.

I. INTRODUCTION

The function of image resizer is to adapt the size of image for display or storage. Applications of image resizing can be found at video play-back system, format conversion, multiparty video conference, etc. Initially, this function is only available in software but now the huge image data rate because of high resolution image and multi-windows makes the hardware implementation necessary.

The motivation of this design is to develop an efficient hardware that can process resizing images of multiple windows in real time. Due to constraints of silicon chip area, we hope that limited hardware resource can be adaptively allocated to windows with different sizes based on the criterion of display resolution given to the corresponding windows. Unlike the intuitive pixel/line dropping and duplication or linear interpolation commonly used in some commercial products, this resizer adopts the filtering process that will result in good image quality. This image resizer can process four 320x200 30fps image resources to display on an 800x600 screen.

II. PREVIOUS APPROACHES

The core technique in image resizer is signal resampling. To avoid aliasing in signal resampling[1], the signal bandwidth should be M times smaller than π , where M is the decimation factor. This limit can be ensured by filtering the signal by an ideal low-pass filter with cut-off frequency π/M . For hardware implementations, some intuitive or simplified resampling methods are developed to reduce the computation complexity and hardware cost.

The first method is linear interpolation. Linear interpolation calculates the relative locations of image lines or pixels to generate the resampling output. Extending the linear interpolation, Fant[2] developed a method whose output is the

weighted sum of neighborhood pixels. However, the neighborhood size and the weight assignment will vary between different applications, which limits the applicability of this method. Another method uses the *sinc* function approach[3][4] to get the interpolated signal. The sampled *sinc* function is approximated by a finite length of sequences to reduce the cost in practical implementations. Besides above approaches, anti-aliasing post/pre-filtering is also applied in the interpolation/decimation processing. All these approaches require multipliers to accomplish the resampling task, which increases the hardware complexity.

Other multiplierless digital filter implementation like distributed arithmetic[5] and cascading half-band filters do not meet the requirement. Distributed arithmetic uses lookup table to implement the multiplication. However, the large data rate will constrain the coefficient word length and precision. Also, the lookup table size will limit the available filter coefficients for different resizing ratios. Although direct cascading half-band filter are suitable for reconfigurable utilization, the power-of-two resizing ratio is too rough to meet the user's requirement. The non-integer resizing ratios smaller than three or four are more commonly used by the windows users. The nine half-band filters proposed by [6] can implement the non-integer ratio resizing by an any order filter, $h(i)=1$. However, it requires the upsampling process before downsampling process to obtain the non-integer ratio. This would take extra computations.

III. CASCADED-INTEGRATOR-COMB FILTERS

We use the cascaded-integrator-comb filters (CIC filters)[7][8] as the resizer modules. CIC filters are inherently multiplierless and can achieve different resizing rate simply by changing the switching rate. Any integer rate resizing can be achieved by just one CIC filter stage, and when more than one stage are available the stages can be cascaded to obtain a finer specified filter response. Besides integer resizing rate, the CIC filter has the potential to do non-integer resizing.

A. Analysis of CIC Filters

Fig. 1 shows the basic structure of the CIC decimation and interpolation filters. They consist of N integrator stages operating at the high frequency f_s and N comb stages operating at the lower sampling rate f_s/R where R is the integer resampling rate. Each comb stage has a differential

delay of M samples and R, N, M are the key parameters to control the frequency response.

The system function of a single integrator stage is $H_i(z)=(1-z^{-1})^{-1}$, and the system function of a single comb stage is $H_c(z)=1-z^{-M}$. The sampling rate is controlled by the switch R between the integrator and comb sections. For decimation, the switch subsamples the output of the last integrator stage, which reduces the sampling rate from f_s to f_s/R . For interpolation, the switch oversamples with rate R by padding $R-1$ zeros between consecutive outputs of the comb section. The overall system function of the CIC filter is

$$H(z) = H_i^N(z)H_c^N(z) = \left(\sum_{k=0}^{RM-1} z^{-k} \right)^N$$

, which is equivalent to a cascade of N FIR stages. The filter parameters R, N, M can be chosen to provide the desired passband assignment and cut-off frequency.

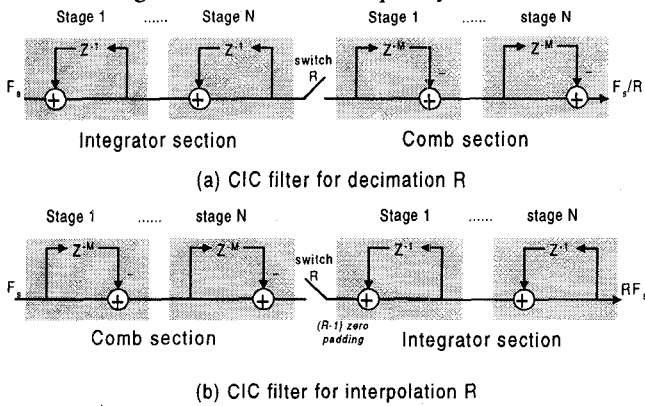


Fig. 1. CIC decimation filter and interpolation filter.

From the system function and architecture of the CIC filters, one can observe the following characteristics of the CIC filters: (1) regular structure and multiplierless, (2) no storage for filter coefficients, (3) wide range of resampling rate R . The only drawback of the CIC filter is the frequency response that has large transition region, passband roll-off and not enough aliasing error rejection in stopband. To eliminate this problem, conventional approach often cascades a FIR filter and a compensator at low sampling rate to shape the frequency response. However, The FIR filter and the compensator will increase the hardware complexity and limit the ability to reconfigure the CIC filters. We desire simple and regular hardware modules for configurable cascading. Our simulation results show that human vision is not sensitive to the imperfect frequency response of CIC filters. Thus, the image resizer is constructed by CIC filter stages only and no extra FIR filter and compensator.

B. Poly-phase Architecture for Non-integer Ratio Resizing

The rate change of conventional CIC filter is limited to be integer. Since the resizing with non-integer ratio is desired in most of the applications, we propose a poly-phase architecture to implement non-integer resizing.

Fig. 2 shows the derivation of 3/2 resizing example. Fig. 2

(a) is a typical rational ratio resizing structure, where $H(z)$ is the CIC filter. Expanding it to its poly-phase structure, we can obtain Fig. 2 (b). Replace $G_1(z)$ and $G_2(z)$ with $(1-Z^{-3})/(1-Z^{-1})$ and rearrange the structure to obtain Fig. 2 (c). To simplify the hardware implementation, we can rearrange the comb and integrator section of Fig. 2(c) separately.

Fig. 3 show the derivation for comb sections of Fig. 2(c). The inputs of B are once cycle delayed due to delay operator before B . So, as Fig. 3(b) shows, after decimation by two, the inputs of $A1$ are $x(0), x(2), x(4) \dots$, the even terms of input sequences and the inputs of $B1$ are $x(1), x(3), x(5) \dots$, the odd terms of input sequences. The two comb stages, $A1$ and $B1$, are the same and operate once every two cycles and the inputs alternate. Hence, we can rearrange Fig. 3 (b) into Fig. 3 (c).

Fig. 4 shows the derivation for integrator sections of Fig. 2 (c). In Fig. 4 (a), the inputs of $A2$ are $0, 0, S_1, \dots$ and the inputs of $B2$ are $0, 0, R_1, \dots$, after zero padding. Due to the delay operator, the inputs of $B3$ are once cycle faster than that of $A3$. The sum of two integrator output in Fig. 4 (b) is equivalent to the result of summing up the two input series and feed to the integrator. So Fig. 4(b) can be simplified to Fig. 4 (c).

Generalizing the above example to any rational resizing ratio, we can obtain the filter architecture of the rational ratio resizing shown in Fig. 5. The system function is

$$H(z) = \frac{1 - Z^{-UD}}{1 - Z^{-\min(U,D)}} (1 + z^{-1} + \dots + z^{-\min(U,D)+1})$$

Except several types of rational resizing ratio (denoted by U/D), most ratio will have narrower passband than it should be according to the resampling algorithms, which will result in some loss of image quality.

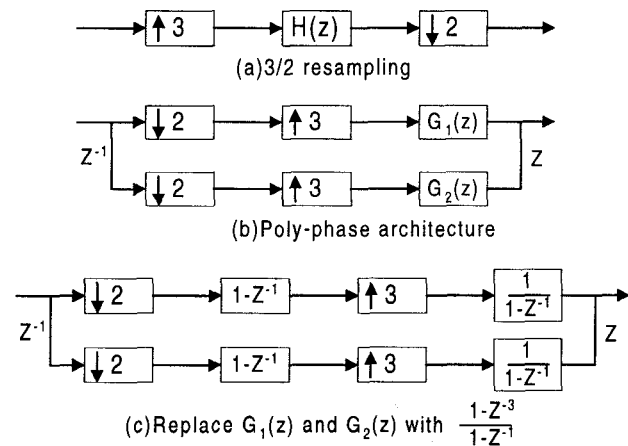


Fig. 2 Development of poly-phase rational ratio resizing.

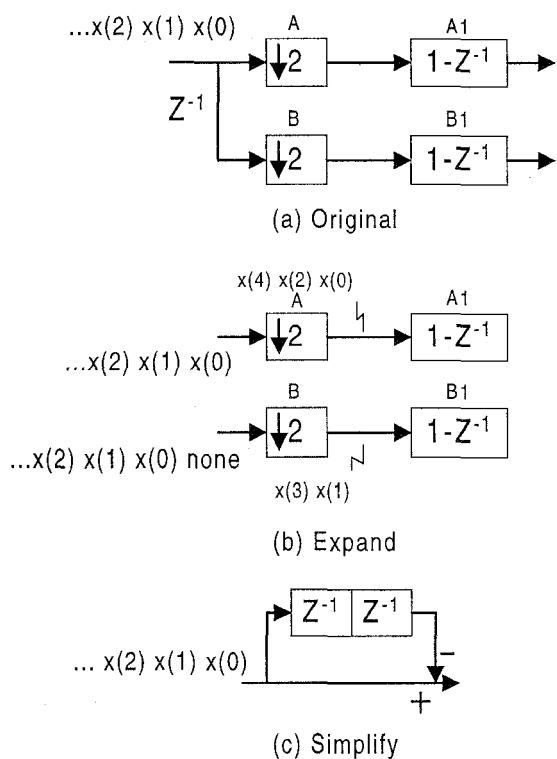


Fig. 3 Derivation of comb sections for rational ratio resizing.

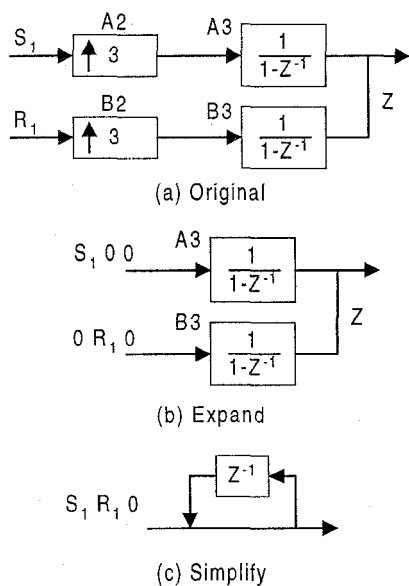


Fig. 4 Derivation of integrator sections for rational ratio resizing.

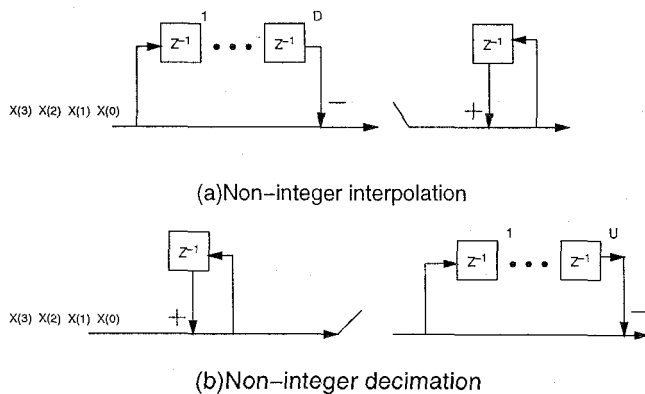


Fig. 5. Implementation of non-integer resizing by CIC filter.

C. Simulation Results of Image Quality

The CIC filter resizing is simulated in C language and the resized images are compared with those produced by Fant's spatial transform technique[2]. From the comparisons, we determined the filter stage allocation and the resizing ratio of the image resizer.

We have simulated three integer ratio resizing interpolation/decimation rates: two, three and four. The simulation results show that:

- (1) one-stage CIC decimation filtering is good enough to produce comparable image quality to that decimated by Fant's method,
- (2) when only one or two CIC filter stages are used in interpolation filtering, the output image has jaggy edges. The more stages are cascaded, the better the image quality is. We decided to use three CIC filter stages, which can generate comparable image quality to that interpolated by Fant's technique.

As to the simulation results of rational ratio resizing, images resized by CIC filters are more blurred than that by Fant's method. This is due to the narrow passband of poly-phase CIC filter, which has filtered out too much high frequency information than desired. Fig. 6 shows the simulation results for 4/3 rational resizing ratio. The test image "Lena" and "spire" are shown here, where the "spire" simulates the effects of high frequency components. Fig. 7 shows the decimated images with different numbers of CIC filter stages.

IV. IMPLEMENTATION ISSUES

A. Allocation of the Filter Stages

The resizer consists of seven CIC filter stages. The host assigns filter stages according to resizing ratio. From previous simulation and analysis, the resizer is designed to implement the following resizing rate: 1/3, 1/2, 2/3, 3/2, 2, and 3.

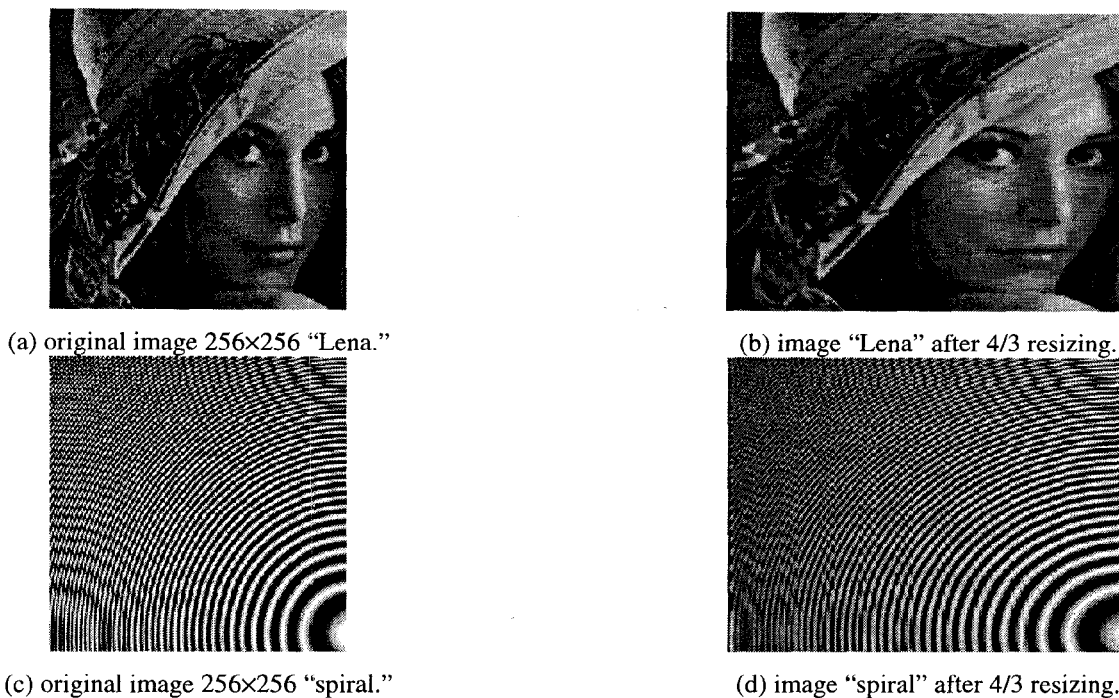


Fig. 6 Image examples resized by 4/3 using 1-stage CIC interpolation filter.



Fig. 7 Original image "Lena" and the resized images by 2/3 (1 stage CIC filter) and 1/2 (2 stage CIC filters), from left to right.

For integer interpolations, more than one stage of filter is desired if the resources are available. At most three CIC filter stages can be assigned to support one interpolation process. For non-integer rate interpolation, only one filter stage filtering is allowed because the passband attenuation increases when the stage number increases. Since more stages of CIC filter make the non-integer ratio resized image more blurred, the 3/2 interpolation uses at most one CIC filter stage. All the image decimations are implemented by one stage CIC decimation filters.

B. Gain of the Resizer

The integer resizing filter gain can be derived from the system function. The system function of one stage 2/3 and 3/2 resizing filter is

$$H(z) = \frac{1 - z^{-6}}{1 - z^{-2}} (1 + z^{-1})$$

$$= 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5}$$

Because of 2/3 resizing's zero padding by two, the effective gain at the filtered data is 6/2=3. Similarly because of zero padding by three, the output gain of 3/2 resizing is 6/3.

Table 1 lists the gains of the CIC filters for different resizing ratio and filter stages. According to the gain list, the maximum register growth will be $\lceil 3 \log_2 3 \rceil$ and the register length shall be $9 + \lceil 3 \log_2 3 \rceil$ bits, that is, 13 bits, to satisfy the two's complement wrap around condition. We do not do any further rounding or truncation to reduce filter register length further. The gain must be scaled down before the output is piped out of the resizer. If the gain is 2 or 4, the scaling down can be simply implemented by bits switch. However, when the gain is not power of two, (for example three or nine), a

particular approach shall be applied to do the scaling down without multiplication. We approach scaling down by 3 by multiplying 3/8 and scaling down by 9 by multiplying 1/8. Scaling down by 8 is done by bits switch, and multiplying 3/8 is done by

$$H'(z) = (H(z) - \frac{1}{2}H(z)) \frac{1}{2}$$

Table 1. Gain of the CIC filters for different resizing ratios.

Resizing ratio	Stage number	Gain
1/3	1	3
1/2	1	2
2/3	1	3
3/2	1	2
2	1	1
2	2	2
2	3	3
3	1	1
3	2	3
3	3	9

C. Memory Reduction by Overlap-Save Block-in

This resizer is designed to use two-pass method to implement 2-D filtering. After the horizontal direction filtering, the filtered data shall be reserved in the delay line to wait for the second filtering process, the vertical direction filtering. The length of delay line is equal to the maximum size of the filtered row length. The size of delay lines required depends on the length of FIR filter.

For the design of CIC filter, the number of delay line depends on the stages numbers. For the input order by scan-in, each delay line unit requires three register data storages and the length of delay line is the maximum horizontal size, 960, where the length of register is derived according to the gain of the resizer. Because four image resources are processed concurrently, four delay lines are required in the worst case when four windows are displayed in the maximum screen width and one fourth of screen height. Such large delay lines will result in high hardware cost in the IC implementation.

To reduce the memory cost of delay line implementation, we arrange the data input in block-in order according to the concept of overlap-save technique. Fig. 8 shows the overlap-save technique. The input is divided to be sections of length N. Each input section overlaps M-1 data with adjacent sections. The conv(*,h) is the convolution result of input section and filter. Discarding the first and the last M-1 section convolution results and cascading the rest results, we get the convolution result that is the same as the convolution of input and filter. We divide the input into N x N blocks according to the overlap-save method. Each input block is filtered row by row then column by column in overlap-save method. An internal buffer of size 3N² bytes are required for each resizing

process.

Mis-selection of overlap-save section convolution occurs when the interval between two successive decimation by R outputs is not R. As shown in Fig. 9, the dash lines mark the down-sample points of the section convolution results. After discarding the first and last M-1 data of section convolution results, mis-selection occurs when the interval between last downsampled point of Sec.A and the first downsampled point of Sec.B is not R.

After discarding the first and last M-1 data of section convolution results, the length of section convolution result is N-(M-1). To avoid the mis-selection of section convolution results for decimation filtering, N-(M-1) must be multiple of decimation rate, R. Hence, the section length N is limited to be 6n+5, where n is non-negative integer. Among the allowed block size, we make a trade-off between buffer size and operation frequency and choose the section length to be 11. Table 2 lists the comparison between block-in and traditional scan-in on frequency and memory cost.

Table 2. Comparisons Between Block-in and Scan-in on Processing Frequency and Memory Requirement

Strategy	Scan-in	Block-in
Frequency	17.28MHz	34.49MHz
Memory Size	149760bits	11616bits

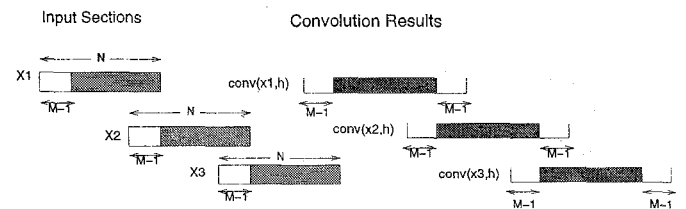


Fig. 8. Block input using overlap-save scheme.

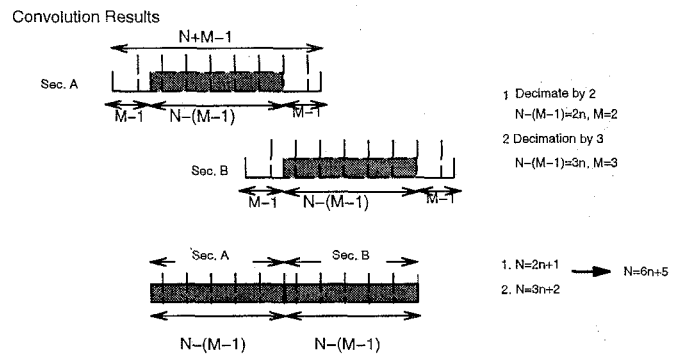


Fig. 9. Mis-selection of Decimation by R.

D. Control Scheme by Concurrent Register Reset

Because filtering is done section by section, the filter register shall be reset to zero when a new section is begun. To reduce the control cost, we propose and compare two register reset strategies, sequential register reset and concurrent register reset.

Fig. 10 shows the sequential register reset strategy. There is a reset control signal path that is parallel to the filter pipeline. At the end of input section A, the reset signal becomes low, the first register in the pipeline is reset. The reset signal propagates along the pipeline and resets the registers stage by stage sequentially. The second method is to reset all filter registers concurrently. Fig. 11 shows the concurrent register reset strategy. When the last input of section A, A2, is fed into the filter pipeline, the reset control signal does not become low until A2 has run through the pipeline. The interval between A2 input and filter output of A2 is the latency of the filter pipeline. After the latency, all registers of the pipeline are cleared together and the next input section B begins. The implementation costs of the two reset strategies are listed at Table 3. The concurrent register reset strategy is adopted in our VLSI implementation.

Table 3. Processing Frequencies When Take Register Clearing Step into Consideration

Clear Strategy	Frequency	Reset Control Cost
Sequential register reset	47.1MHz	3172.4(gate count)
Concurrent register reset	53.4MHz	215.2(gate count)

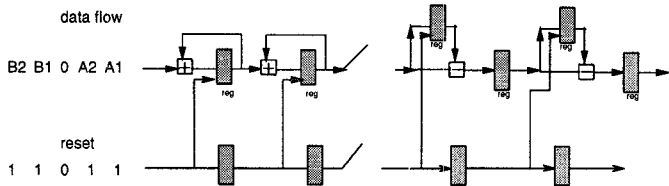


Fig. 10. Sequential Register Reset.

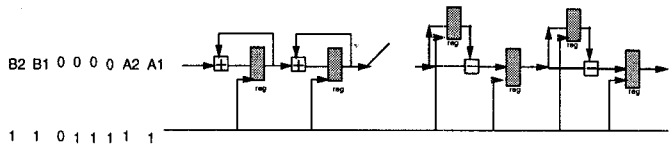


Fig. 11. Concurrent register cost.

V. CHIP DESIGN

The input images of the resizer IC can be up to four 320x200 8-bit image sources with frame rate 30 frames per second. The output display size is 800x600 and the resizing ratio range from 3 to 1/3. The gate count of the resizer chip is about 40K in which four 512x8 SRAMs are included, and the die size is 7076µm×7429µm. This chip is designed and implemented by using ITRI/CCL 0.8µm SPDM CMOS cell library. The floorplan is shown in Fig. 12. The Verilog simulated results show that this chip can run up to 55.55Mhz.

The functions of the main components in the resizer includes:

1. **Filter Block:** This block includes **Filter Set** and **Post**

Process. The **Filter Set** is constituted by seven configurable CIC filter stages and switches shown in Fig. 13. The two multiplexers *IN_Idmux* and *feed_out* controls the data flow between the comb stages and integrator stages for interpolation filtering and decimation filtering. The **Post Process** takes charge of scaling down the filter gain.

2. **Filter reconfiguration controller:** We use a 10x22 PLA to generate the resizing control signals for *Filter Set*. The host shall send the resizing rate code and filter configuration signals to this controller when a new frame begins.
3. **Controller:** The controller generates the signals to reset the filter registers at the end of each input section. The read of block-in and write of overlap-save convolution result are also controlled according the control data read from this controller.
4. **Internal Buffer:** Four 363-byte on-chip SRAMs are used for intermediate data of two-pass filtering.

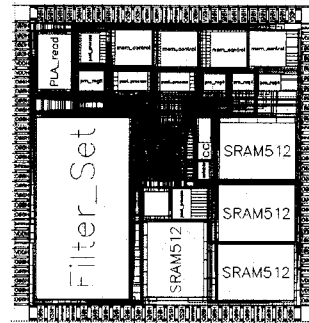


Fig. 12. Floorplan of the resizer chip.

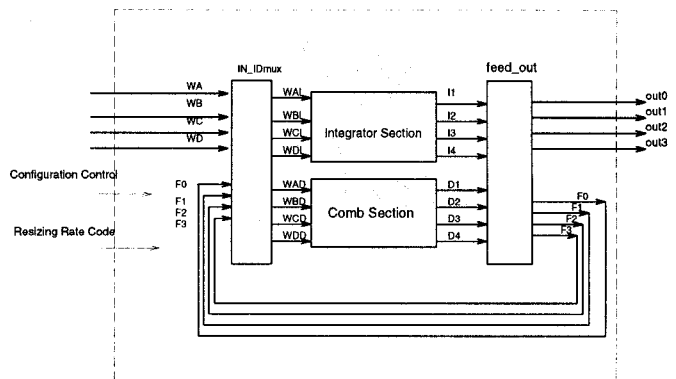


Fig. 13. Block Diagram of the *Filter Set*.

VI. CONCLUSIONS

In this paper, an image resizer has been designed for multiple-window displays. It featured the simple reconfigurable filter stages that can be dynamically allocated to the image windows with different size. The resizing quality is acceptable in real-time processing. The silicon area has been greatly reduced by using overlap-save based block-input

and concurrent register reset. The Verilog description of this IC design has been verified. The simulated results shows that the resizer can run at 55.55Mhz.

REFERENCES

- [1] A. V. Oppenheim and R.W. Schaffer, *Discrete Time Signal Processing*, Prentice-Hall Inc., 1989.
- [2] K. M. Fant, "A nonaliasing, real-time spatial transform technique," *IEEE Computer Graphics and Applications*, p. 71, Jan. 1986.
- [3] S. Kim and W. Su, "Direct image resampling using block transform coefficients," *Signal Processing: Image Communications*, vol. 5., May 1993.
- [4] L. Capodiferro, A. Chiari, G. Marcone, and S. Miceli, "A screen format converter for HDTV," *Signal Proceeding of HDTV, III*, 1992.
- [5] A. Peled and B. Liu, "New hardware realizations of digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-22, pp.456-462, Dec. 1974.
- [6] D. J. Goodman and M.J. Carey, "Nine digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-25, p. 121, April 1977.
- [7] S. Chu and C.S. Burrus, "Multirate filter designs using comb filters," *IEEE Trans. Circuits & Syst.*, vol. CAS-31, p. 913, Nov. 1984.
- [8] E. B. Hogenaues, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. ASSP*, vol. 29, pp. 155-162, Apr. 1981.

Biographies



Ching-Mei Huang received the B.S. and M.S. degree in electronics engineering from National Chiao Tung University in 1993 and 1995, Hsinchu, Taiwan.

She is currently with Multimedia department of Silicon Integrated Systems Corporation, Hsinchu, Taiwan. Her research interests include digital filter designs, multimedia signal processing and digital signal processing.



Tian-Sheuan Chang (S'93-) received the B.S. and M.S. degree in electronics engineering from National Chiao-Tung University in 1993 and 1995, Hsinchu, Taiwan.

He is currently working on the Ph.D. degree in electronics engineering at National Chiao-Tung University. His research interest includes VLSI design, digital signal processing and computer architecture.



Chein-Wei Jen (S'78-M'87) received the B.S. degree from National Chiao Tung University in 1970, the M.S. degree from Stanford University, Stanford, CA, in 1977, and the Ph.D. degree from National Chiao Tung University in 1983.

He is currently with the Department of Electronics Engineering and the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, as a Professor. During 1985-1986, he was with the University of Southern California, Los Angeles, as a Visiting Researcher. His current research interests include VLSI design, digital signal processing, processor architecture, and design automation. He has held four patents and published over 30 journal papers and 70 conference papers in these areas.

Dr. Jen is a member of the honor society Phi Tau Phi. He received the 1990 Best Paper Award from the Engineer Society, the 1989-1996 Long-Term Best Paper Awards from Acer, the 1994, 1995 Best Paper Awards of the HD-media conference. He was the Program Committee member of ICCD'94, ICCE'95-97. He is currently an editor of the *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*.