Reduction of Power in Phase unwrapping algorithm using RCA

Fathima Beevi M[1], Saju A[2] and Vishnu Raj[3]

[1]PG Scholar, Musaliar College of Engineering and Technology,
Pathanamthitta, Kerala, India

[2]Associate Professor, Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India

[3]Assistant Professor, Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India

Abstract. The paper presents a phase unwrapping architecture for imaging applications. The architecture is that theimplementation of a path-independent non-iterative Discrete Cosine Transform (DCT) based minimum mean square algorithm for accurate and fast phase unwrapping. The implementation is based on field programmable gate array. The architecture is able to exploit the parallelism among different stages of the algorithm for maximizing the throughput of the computation. To reduce the power a Ripple Carry Adder (RCA) is used. As compared with other implementations for fast phase unwrapping, the proposed architecture has the advantages of high throughput, high accuracy, and low power consumption. It is designed using Verilog HDL and is implemented using Xilinx 14.2 ISE tools.

Keywords: Discrete Cosine Transform, Field Programmable Gate Array, Phase Unwrapping, Ripple Carry Adder.

1 Introduction

In the advanced imaging technologies such as Digital Holographic Microscopy (DHM) [1], Magnetic Resonance Imaging (MRI) [2] and Synthetic Aperture Radar (SAR) [6], phase measurement and processing are required. The measured phase is normally wrapped between the values of $-\pi$ and π, resulting in phase discontinuities. Phase unwrapping [14] operations are usually desired to remove the discontinuities for the subsequent phase processing or image rendering. To employ a phase unwrapping algorithm, the throughput, power and area complexities are usually the important concerns. A number of phase unwrapping algorithms have been proposed to improve the accuracy at the expense of higher computation complexities. The Goldstein's algorithm (4) improves the noise immunity by a branch cut technique. The noise removal can also be carried out by the windowed Fourier transform filter before phase unwrapping. The region reference techniques [7] have been found to be effective for phase discontinuity detection. The Preconditioned Conjugate Gradient (PCG) [11], Gauss-Seidel technique, Discrete Cosine Transform (DCT) [13], or Successive Over Relaxation (SOR) [5] performs the phase unwrapping based on the minimum mean square criterion. The goal of these methods is to obtain an unwrapped solution by minimizing the differences between the derivatives of the solution and those of the wrapped phase measurements. The Accumulation of Residual Maps (ARM) [9] is a variation of the minimum mean square algorithm providing a multigrid approach for efficient iterative computation. A common drawback of some existing phase unwrapping algorithms is that they are iterative. In addition to having higher computational complexities, the number of iterations would be dependent on the content of the unwrapped phases. Consequently, the computation time would be content dependent. In addition, for some iterative algorithms, the convergence of the iterations may not be guaranteed. This may not be advantageous for applications such as DHM [1], where a fixed and high throughput is desired for real time Three Dimensional (3D) imaging. Field Programmable Logic Array (FPGA) [3] devices may consume less power. Therefore, it may be difficult for these architectures to provide fixed and high throughput.

2 Phase unwraping algorithm

The method offers efficient pipeline operations for the DCT-based minimum mean square algorithm. The architecture can be separated into four stages: Laplacian operations, modified DCT, frequency domain operations, and Inverse DCT (IDCT). The circuit is able to fully exploit the parallelism among these stages so that the partial results produced by each stage can be immediately processed by the subsequent ones. The feature is benificial for minimizing the number of memory accesses to the intermediate results. It is also helpful to further reduce the power, area and increases the throughput for the phase unwrapping.

Let $\psi i, j$ be a wrapped phase of an original phase $\beta i, j$ for $0 \le i, j \le N - 1$, where N is the size of the rows and columns of the phase planes. The $\psi i, j$ *and* $\beta i, j$ is related by;

$$\beta i, j = \psi i, j + 2\pi k i, j \text{----------(1)}$$

Where ki, j is an integer, and $-\pi \le \psi i, j < \pi$. Define the phase differences $\Delta^x i, \ j$ and $\Delta^y i,j$ as

$$\Delta^x i, j = W(\psi i+1, j - \psi i, j) \text{ ---------- (2)}$$

$$\Delta^y i, j = W(\psi i, j+1 - \psi i, j) \text{ ---------- (3)}$$

Where the function W denotes as a wrapping operator that wraps all values of its argument into the range $[-\pi, \pi)$ by adding or subtracting an integral number of 2π from its argument. Therefore, $\Delta^x i, j$ and $\Delta^y i, j$ is in the range of $-\pi$ to π. The goal of the DCT-based phase unwrapping algorithm is to find $\beta i, j$ minimizing the following cost function J:

$$J = \sum_{i=0}^{N-2} \sum_{j=0}^{N-1} (\alpha_{i+1,j} - \alpha_{i,j} - \Delta^x i, j)^2 + \sum_{i=0}^{N-1} \sum_{j=0}^{N-2} (\alpha_{i,j+1} - \alpha_{i,j} - \Delta^y i, j)^2 \text{----(4)}$$

It can be shown that the following discrete Poisson's equation [4], [8] can be used for minimizing J:

$$\varphi i+1, j + \varphi i-1, j + \varphi i, j+1 + \varphi i, j-1 - 4\varphi i, j = \gamma i, j \text{------(5)}$$

Table I. Noniterative DCT-based phase unwrapping algorithm steps.

Require: Boundary conditions in (7) (8).
1. Compute Δ^x i, j by (2), $0 \le$ i $\le$ N-2, $0 \le$ j $\le$N-1.
2. Compute Δ^y i, j by (3), $0 \le$ i $\le$ N-1, $0 \le$ j $\le$N-2.
3. Compute γi, j by (6), $0 \le$ I,j $\le$ N-1.
4. Compute ϕ by (9).

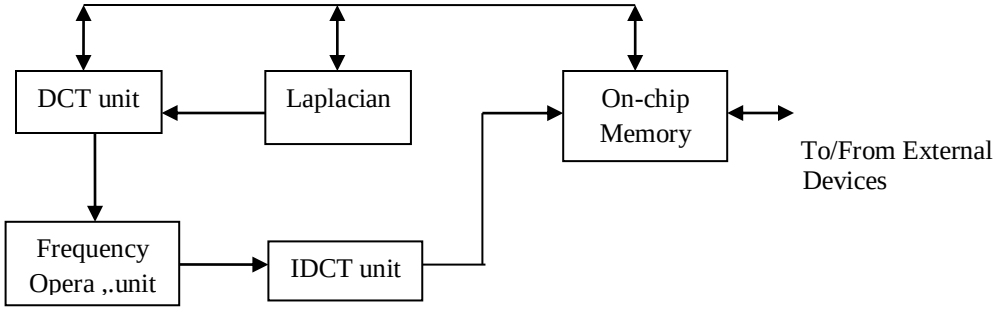

Fig. 1. The architecture for phase unwrapping

Where

$$\gamma i, j = \Delta^x i, j - \Delta^x i-1, j + \Delta^y i, j - \Delta^y i, j-1 \text{---------- (6)}$$

The $\gamma = \{\gamma i, j: 0 \le i, j \le N - 1\}$ can be viewed as the results of Laplacian operations over ψ with phase wrapping.

Based on the boundary conditions

$$\Delta^x -1, j = 0, \Delta^x N - 1, j = 0, j = 0, ..., N - 1 \text{----------(7)}$$

$$\Delta^y i, -1 = 0, \Delta^y i, N - 1 = 0, i = 0... N - 1 \text{---------- (8)}$$

$\beta i, j$ in (5) can be solved by DCT [4]. Let α and Γ be the $N \times N$ DCT of β and γ, respectively. The solution to (5) can then be first computed in the DCT domain as

$$\alpha m, n = \Gamma m, n 2 \cos(\pi m/N) + 2\cos(\pi n/N) - 4 \text{---------- (9)}$$

For $0 \le m, n \le N - 1$. The 2D IDCT is then applied to α to obtain the unwrapped phase β, with mirror reflection and scaling, the FFT and Inverse FFT (IFFT) can be used for the computation of DCT and IDCT. The employment of FFT and IFFT would be beneficial for reducing the computation complexities of the DCT-based phase unwrapping algorithm.

3 Architecture of phase unwrapping

The architecture can be separated into six units: the Laplacian unit, the Modified DCT unit, the frequency domain operation unit, the IDCT unit, and the on-chip memory. The Laplacian unit fetches $\psi i, j$ from the on-chip memory, and computes ($\gamma i, j$). The DCT unit computes the DCT of γ and produces Γ. The goal of frequency domain operation unit is to computes Γ and produce α. The IDCT unit then finds the unwrapped solution β by carrying out the IDCT on α. The on-chip memory holds the input phase image ψ, the intermediate results produced by the other units, and the final unwrapped phase image β. The on-chip memory also acts as the buffer for the external accesses.

3.1 Laplacian Unit

The Laplacian unit fetches pixels of ψ one at a time in raster scan order from the on-chip memory, and produces pixels of γ one at a time in the same order to the on-chip memory. The circuit contains an Address Generation Unit (AGU), a counter, two shift registers, two phase wrapping units, and a number of adders. The AGU is responsible for producing addresses for reading $\psi i, j$ and writing $\gamma i, j$ from/to on-chip memory. The remaining parts of the circuit are used for Laplacian operations. Without loss of generality, assume the circuit is currently producing $\gamma i, j$. The values of the indices i and j are determined by the counter of the circuit supporting the raster scan ordering. From (6) $\Delta^x i, j$, $\Delta^x i-1, j$, $\Delta^y i, j$, and $\Delta^y i, j-1$ are then required. Two phase wrapping modules are adopted for the computation of $\Delta^x i, j$ and $\Delta^y i, j$.

The architecture of the phase wrapping modules, which support the comparisons, addition/substraction, boundary detection, and multiplexing operations. The comparators and adders/substractors are used for wrapping the input values to the range of $(-\pi, \pi)$. The comparators are used to determine the interval the input belongs to based on the set of thresholds $\{(2p + 1)\pi, p = -q... q\}$. Define the interval $Ip = [(2p - 1)\pi, (2p + 1)\pi)]$ for $p = -q... q$. When input belongs to Ip, it will be added by $-2p\pi$ for phase wrapping by the corresponding adder and multiplexer. For our applications, because $-\pi \le \psi i, j < \pi$, it is clear that the differences between two pixels in ψ would be in the range of $-2\pi$and 2π. Therefore, the selection of $q = 1$ is sufficient to cover the range of input values for phase wrapping.

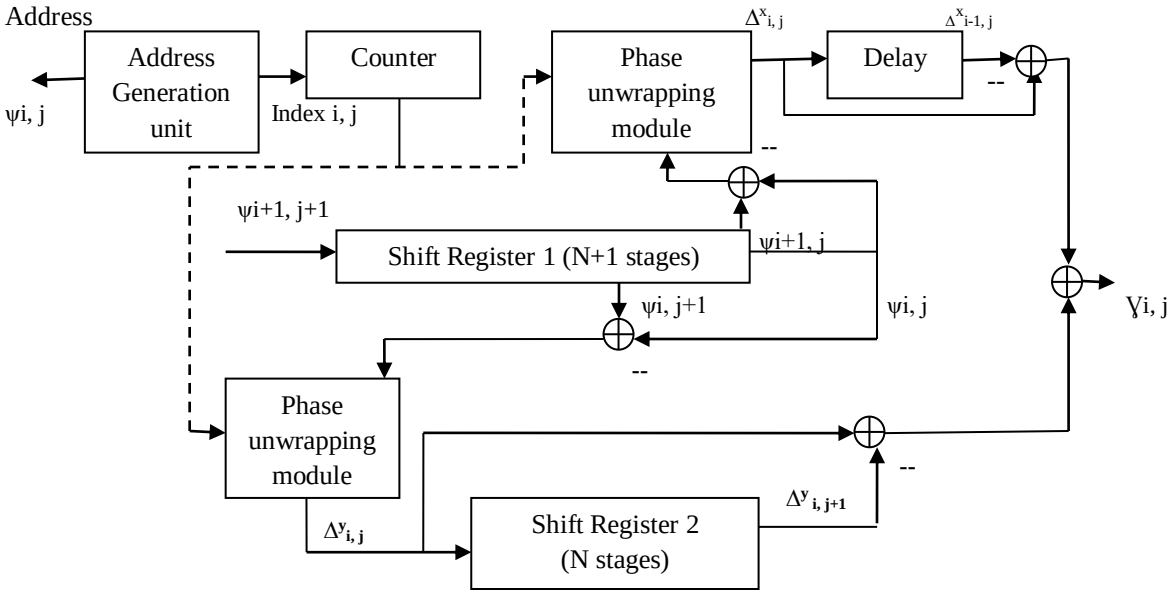

Fig. 2. The architecture of Laplacian Unit

The boundary detector is used for the enforcement of the boundary conditions stated in (7) and (8). It is based on a simple rule that the boundary detection would be true when $i = 0$ or $(N - 1)$, and/or $j = 0$ or $N - 1$. In these cases, the output of the module is zero. Otherwise, the output would be the result of the wrapping operations.

3.2 DCT Unit

The DCT unit brings an explanation of the 2D DCT over γ composed by the Laplacian unit. The 2D DCT with size N × N is separable and can be achieved by 1D DCT with size N. Moreover, using mirror reflection and scaling, we can achieve 1D DCT with size N by 1D FFT with size 2N. To develop on the operations, the number of 1D DCT by 1D FFT is first examined. Let s = {si, i = 1... N} is a 1D sequence. Also, let S be the DCT of s. They are described by:

$$Sm = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} bmsi \cos\frac{\pi(2i+1)m}{2N}, m = 0, \dots, N-1 \text{---------- (10)}$$

Where

$$bm = \begin{cases} \frac{1}{\sqrt{2}} & m = 0, \\ 1 & otherwise. \end{cases} \text{ ---------- (11)}$$

Define $\widehat{si}$, i=0… 2N-1, as

$$\widehat{Si} = \begin{cases} si & 0 \le i \le N-1, \\ s2N-1-i & N \le i \le 2N-1 \end{cases} \text{ ---------- (12)}$$

We can see that ˆ*s* as the extension of *s* by mirror reflection. Let ˆ *S* be the FFT of ˆ*s*. That is,

$$\hat{S}m = \sum_{i=0}^{2N-1} \widehat{si}\ W\frac{-im}{2N}, m = 0, \dots, 2N-1 \text{ ---------- (13)}$$

Where complex exponential *WN* = exp (***j*** 2*πN)*. It can be derived from (10), (12), and (13) that

$$Sm = \frac{1}{\sqrt{2N}} bmW_{2N}^{-m/2} \hat{S}m, , m = 0, \dots, N-1 \text{ ---------- (14)}$$

Therefore, with the mirror and scaling operations given in (12)–(14), *S* can be obtained from *s* by FFT. These result can be effectively applied to the 2D DCT over *γ*, as shown below:

$$\Gamma m,n = \frac{2}{N}\sum_{j=0}^{N-1}\sum_{i=0}^{N-1} bm\ bn\gamma i,jcos\frac{\pi(2i+1)m}{2N}cos\frac{\pi(2j+1)n}{2N} \text{---------- (15)}$$

Observe that (15) can be rewritten as

$$\Gamma m,n = \sqrt{\frac{2}{N}}\sum_{j=0}^{N-1} bn\Lambda m,jcos\frac{\pi(2j+1)n}{2N}, \text{ ---------- (16)}$$

$$\text{Where}\Lambda m,j = \sqrt{\frac{2}{N}}\sum_{i=0}^{N-1} bn\gamma i,jcos\frac{\pi(2i+1)m}{2N} \text{ ---------- (17)}$$

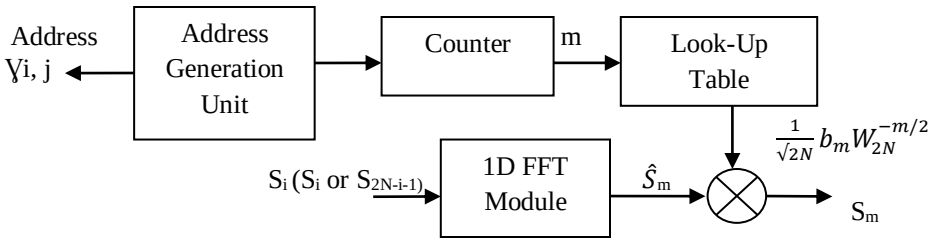

Fig. 3. The architecture of DCT Unit

3.2.1 Ripple Carry Adder

The Ripple Carry Adder is assembled by cascading Full Adders (FA) blocks in sequence. One Full Adder is held for the addition of two binary digits at any step of the Ripple Carry. The carryout of one step is served directly to the carry-in of the next stage. Also though this is a simple adder and can be used to add allowable bit length numbers, it is still not very useful when large bit numbers are used. One of the most serious disadvantages of this adder is that the delay increases linearly with the bit length. The worst-case delay of the RCA is when a carry signal transformation ripples through all stages of the adder chain from the least significant bit to the most significant bit.

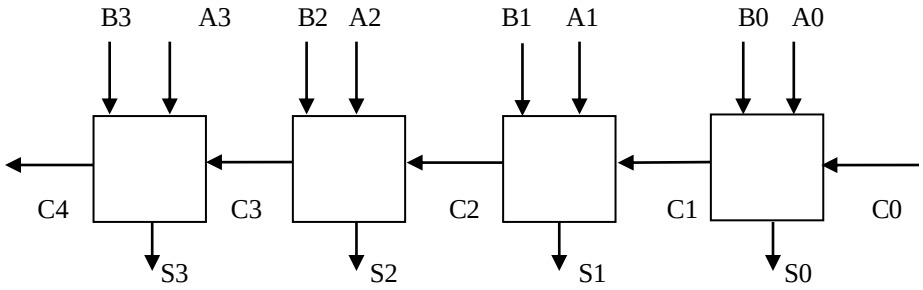

Fig. 4. Ripple carry adder

To use the 1D FFT for the 2D DCT, rows of γ are the first estimated one at a time using 1D FFT. By connecting (10) with (17), we can see that s is a row of γ for the row-wise sum. The resulting S is then the corresponding row of Λ. After the Λ is concerned, columns of Λ are then computed one at a time. Of (10)–(16), the s can be a column of Λ for the column-wise computation. The resulting S is the corresponding column of Γ. Behind the 1D DCT of all the columns of Λ are made, the Γ is then obtained. Fig. 3 shows the architecture of the DCT unit, which consists of an AGU, a 1D FFT module, a Look-Up Table (LUT), a multiplier, a counter, and a controller. In addition to producing addresses for memory accesses, the AGU is responsible for the mirror reflection operation shown in (12) for

retrieving ˆs from memory. This can be achieved by first producing the addresses for obtaining s, and then the addresses for the symmetric addition of s. Because the size of ˆs is 2N, the transform length of the FFT is 2N. The FFT module includes [log4 (2N)] stages. Each stage has a single butterfly. The 1D-FFT module has a single data input and single data output. It would then takes 2N clock cycles to fetch inputs from the input buffer. Because the module has only a single output; it would also take 2N clock cycles to collect all the outputs from the module. As pointed in (14), only the first N outputs will be used by the scaling operations for the number of Sm, m = 0... N − 1. They will be sent back to on-chip memory for consequent operations. The last N outputs will be discarded. The scaling operations are performed by the LUT, the multiplier and the counter is shown in fig. 3. The LUT contains N entries, where the value $\frac{1}{\sqrt{2N}} b_m W_{2N}^{-m/2}$ is stored in the entry m, m = 0... N − 1, of the LUT. The counter remains the index m of ˆ Sm produced by the module. The multiplier is able to carry out the complex multiplication with floating point format. Related to the adders in the Laplacian unit, it supports multiple-clock fully-pipelined addition. The high 1D DCT operations can be improved to 2D DCT operations by row-wise computation over γ shown in (17), and column-wise computation over Λ reported in (16). Because the 1D FFT modules is fully pipelined, continuous rows (or columns) can be seamlessly loaded to the module, for row-wise operations. The loading and writing operations for each row takes 2N and N clock cycles, respectively. Let L2 be the latency of the FFT and scaling operations. We then see from Fig. 7 that the total computation time for row-wise operations is 2N2 + L2 + N clock cycles. Likewise, the total computation time for column-wise operations is 2N2 + L2 + N clock cycles. The total computation time for the 2D DCT is then 2(2N2 + L2 + N) clock cycles.

3.3 Frequency Operation Unit

Given Λ generated by this DCT unit, the frequency operation unit computes α by (9). The unit implements Step 5 of the algorithm of non-iterative DCT based phase unwrapping. Comparatively presented in Fig. 5, the unit includes an AGU, a counter, two LUTs, an adder, and a divider. Comparable to the preceding two units, the frequency operation unit fetches pixels from on-chip memory one at a time. The fetching process is identical to the column-wise FFT operations in the DCT unit. This fetching system may be helpful for overlapping the operations of DCT unit, frequency operation unit and IDCT unit. The core part of the circuit is to compute the denominator of (9). Here is performed by table lookup and addition. Separate The LUT of the circuit contains N entries. The entry m of the LUT1 and LUT 2 contain the values of 2 cos πm/N and 2 cos πm/N− 4, sequentially. It can be observed from Fig. 5 that the counter follows the indices m and n of the input Γm, n. The contents are used as the inputs to the LUTs. The Γm, n is then divided by the sum of the output of the LUTs. Both the adder and the divider are fully pipelined for the floating-point operations. Make L3 be the number of clocks needed to compute output αm, n from input Λm, n. The whole estimate time for the frequency operation unit is then N2 + L3 clock cycles.

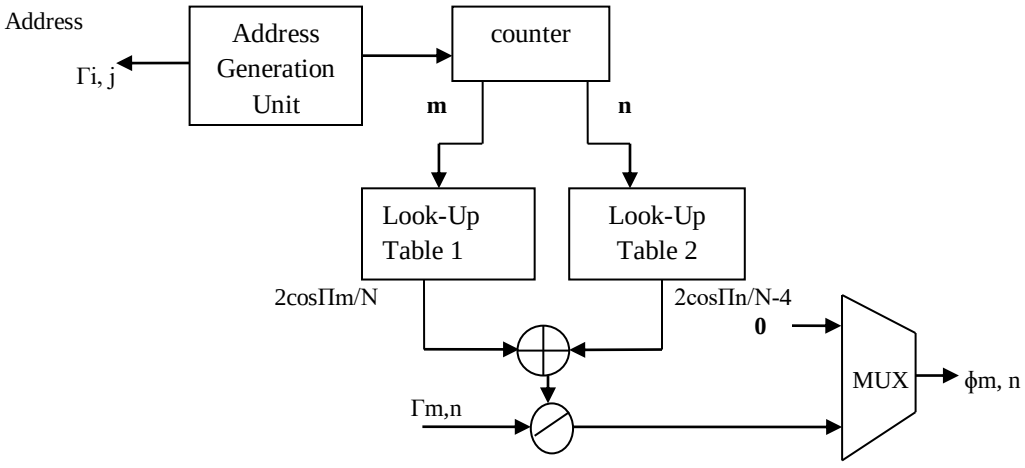

Fig. 5. The architecture of Frequency Operation Unit

3.4 IDCT Unit

The object of the IDCT unit is to complete IDCT covering α to get the unwrapped phase β, as shown below:

$$\alpha i,\ j=\frac{2}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} b_m b_n \beta_{m,n}\ cos\frac{\pi(2i+1)m}{2N}\ cos\frac{\pi(2j+1)n}{2N} \text{--------- (18)}$$

Following a related method to the 2D DCT case presented in (15)–(17), us first assume that the separability of 2D IDCT provides the 2D IDCT to be taken out by column-wise and row-wise1D IDCT calculations. Suppose succeeding the resulting 1D IDCT with size N and IFFT with size 2N:

$$s_i=\sqrt{\frac{2}{N}}\sum_{m=0}^{N-1} b_m s_m\ cos\frac{\pi(2i+1)m}{2N}, i=0\ldots N-1 \quad \text{---------- (19)}$$

$$\widehat{s_i}=\frac{1}{2N}\sum_{m=0}^{2N-1}\widehat{S_m}\ W_{2N}^{im}, i=0,\ldots\ldots 2N-1 \quad \text{---------- (20)}$$

It can be given by

$$\widehat{S_m}=\begin{cases}\sqrt{\frac{2N}{bm}}W_{2N}^{0.5m} & 0\le m\le n-1\\ 0, & m=n\\ -\sqrt{\frac{2N}{bm}}W_{2N}^{0.5m}S_{2N-m,}N+1\le m\le 2N-1\end{cases} \quad \text{---------- (21)}$$

Then

$$s_i=\widehat{s_i}, i=0,\ldots\ldots,N-1 \quad \text{---------- (22)}$$

Address

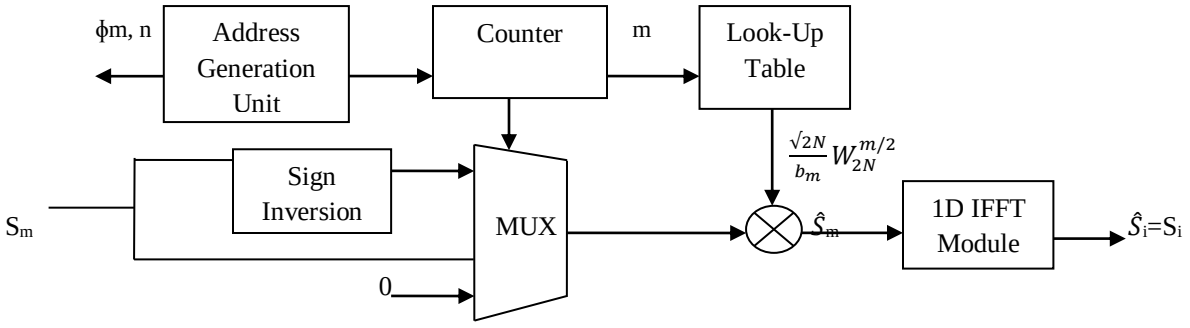

Fig. 6. The architecture of IDCT Unit

Consequently, including the mirror reflection and scaling operations shown in (21), 1D IDCT in (19) can be achieved by 1D IFFT in (20). Accordingly, related to the DCT unit, the 2D IDCT with size N × N can be completed by the 1D IFFT with size 2N.

Because 1D FFT and IFFT could like the same circuit module, the DCT unit with changes could be re-used for the IDCT computation while the DCT and IDCT are operated sequentially. But, in the advanced architecture, parallel operations of DCT and IDCT are taken out to succeed more powerful throughput. Hence, as shown in fig. 6, the IDCT unit has its 1D IFFT module for a dynamic estimate. The significant difference between the DCT and IDCT architectures is the LUT-based scaling operation. The scaling operation is completed before the transform in the IDCT unit. By difference, we can see from fig. 3 that it is carried out after the transform in the DCT unit. Also,

besides as exhibited in (21) sign opposite is needed for the mirror reflection for IDCT. A simple sign inversion circuit and a multiplexer are then incorporated for this purpose.

To operate concurrently with the DCT unit, the IDCT unit starts with the column-wise operations. In this way, partial columns DCT issues, later passing through the frequency operation unit, can be directly prepared by the IDCT unit. This enables the efficient exploitation of parallelism among different pixels of the intermediate results of the phase unwrapping.

Following the achievement of column-wise operations, each row of the resulting matrix is then carried to the IDCT module for the row-wise estimate. The final result β is taken after the row-wise computation is finished. Because the IFFT module is completely pipelined, the methods of column-wise and row-wise estimates of the IDCT unit. Let L4 be the latency of the scaling and IFFT operations. Both the row-wise and column-wise operations have the same total counting time 2N2 + L4 + N clock cycles. The total estimated time for the 2D IDCT is then 2(2N2 + L4 + N) clock cycles.

3.5 On-Chip Memory

The on-chip memory is applied for depositing the source, intermediate and resulting data. The purpose of the on-chip memory is dependent on storing the data. One easy coordination program is to achieve a regular program where the Laplacian unit DCT unit, frequency operation unit and IDCT unit are operated continuously. Each unit will not be stimulated until the estimate of the previous unit is closed. Since only one unit is stimulated at a time. Because the on-chip memory doesn't need to hold the pixels of the source data which have been fetched, the areas of the fetched pixels can be reused to store the data generated by the activated unit. Besides, the source data and results have equal size N × N. Accordingly the size of the on-chip memory could be N× N for the single consecutive state.

The major drawback of the following record is the long latency. The total latency for the sequential schedule, denoted by Tseq, is then given by

$$T_{seq} = \sum_{k=1}^{4} T_k = 10N^2 + L_1 + 2L_2 + L_3 + 2L_4 + 5N = 1 \text{---------- (23)}$$

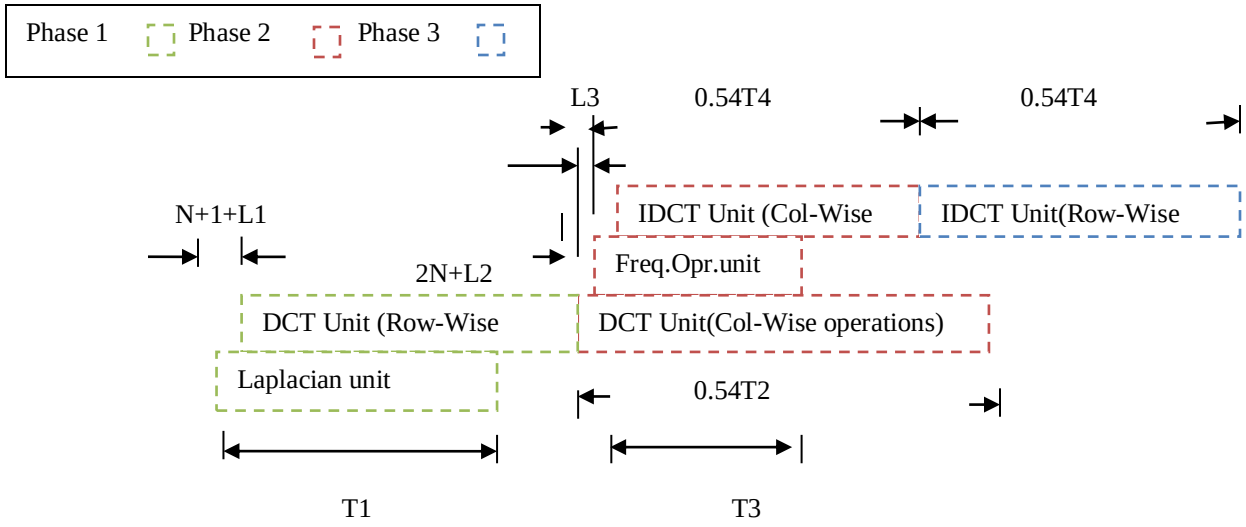

Fig. 7. The Parallel Operations

While the first phase, the row-wise operations of the DCT unit are stimulated directly after the first output pixel of Laplacian unit is provided. It takes N +1+L1 clock cycles for the Laplacian unit to provide the original output, and 0.5T2 clock cycles for the row-wise DCT computation. The latency of phase1 is consequently N +1+L1 + 0.5T2.

Furthermore, as the next phase, the frequency operation unit is initiated after the first output pixel produced by the DCT unit for column-wise operations is available, which takes 2N + L2 clock cycles. Also, it takes L3 clock cycles for the frequency operation unit to produce its first output. Accordingly, the activation of the column-wise operations of the IDCT unit will be L3 clock cycles after the activation of the frequency operation unit. The latency of the next phase is 2N + L2 + L3 + 0.5T4. The row-wise computation of IDCT is subsequently carried out alone during phase three, which has a latency of 0.5T4. Let Tpar be the total latency for the parallel schedule. It can then be observed from Fig. 8 that

$$Tpar = 3N +1+L1 + L2 + L3 + 0.5T2 + T4, = 6N2 + 6N +1+L1 + 2L2 + L3 + 2L4 \text{---------- (24)}$$

For large N, it can be observed from (23) and (24) that Tseq ≈ 10N2, and Tpar ≈ 6N2. The decrease in latency for the parallel schedule as compared with its regular complement is therefore 40%. To accommodate the parallel schedule, there are two RAM modules (denoted by RAM 1 and RAM 2) in the on-chip memory. The size of each RAM module is N × N. Fig. 8 shows the actions of the RAM modules with the Laplacian, DCT, frequency operation, and IDCT units for different phases of the parallel schedule. Throughout phase 1, the Laplacian unit fetches ψ from RAM 1 and writes back γ to RAM 2.

The DCT unit fetches γ from RAM 1 and writes back Λ to RAM 1. The locations of the fetched pixels of ψ inRAM1 are re-used for holding the pixels of Λ produced by the DCT unit. Consequently, there is no structural hazard for the parallel execution of Laplacian and DCT units. While phase 2, the DCT unit reads Λ from RAM 1 and delivers the computation results Γ directly to the frequency operation unit. Because there is no need to carry out the mirror reflection for the frequency operation unit, the data produced by the DCT unit can continue directly to the frequency operation unit without writing back to the on-chip RAM, as depicted in Fig. 8(b). The frequency operation unit then stores its computation results α back to RAM 2. The IDCT unit then fetches α from RAM 2, and writes back to column-wise computation results to RAM 1. Finally, only RAM 1 is involved in the row-wise IDCT computation during phase 3. It more stores the final phase unwrapping results β.

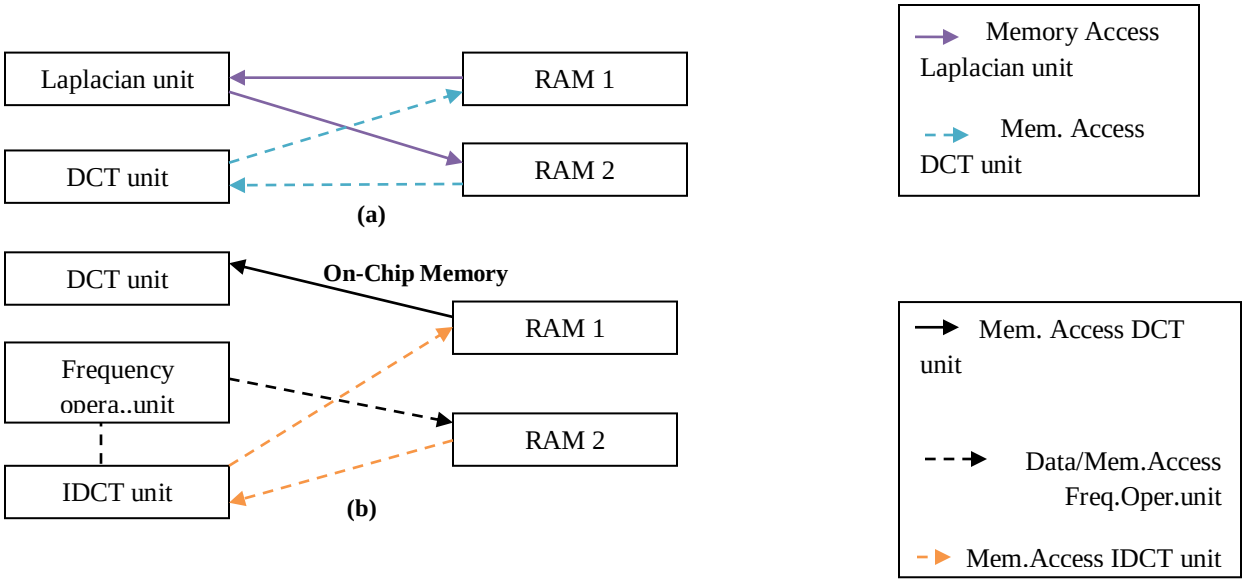

Fig. 8. Operations of the on-chip memory for the parallel schedule: (a) Operations during phase 1 of the schedule, (b) Operations during phase2 of the schedule.

4 Result And Discussion

All the blocks are modeled using Verilog HDL.The simulation results of the proposed design are verified using Xilinx ISE 14.2 and reduce the area and power consumption based on adders, multipliers, dividers and registers are the basic building blocks of this architecture, the area complexities are separated into four categories: the number of adders, the number of multipliers, number of dividers and the number of registers.

The present phase unwrapping is based on different algorithms. Also, these are realized by different platforms. Hence, it may be trying to directly connect the review of these method. But, when the power is an important concern, it can be seen from Table 1 that the advanced architecture is an effective alternative for applications requiring fast unwrapping. the advanced architecture with the parallel schedule has low power consumption over many of the present method.

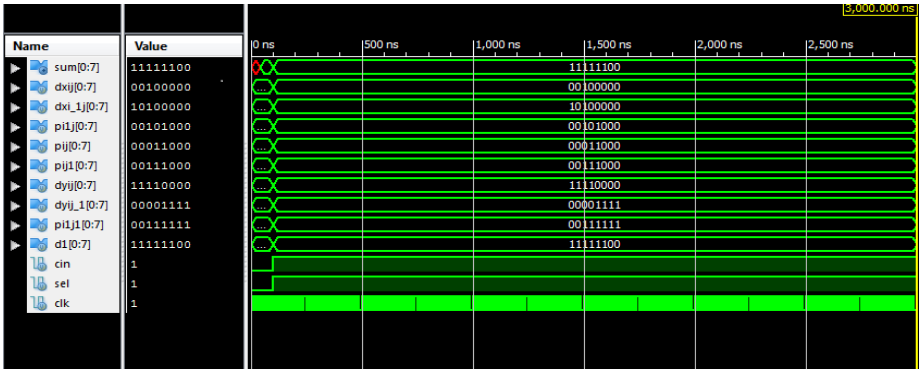

Fig.10. Simulation result of Laplacian unit

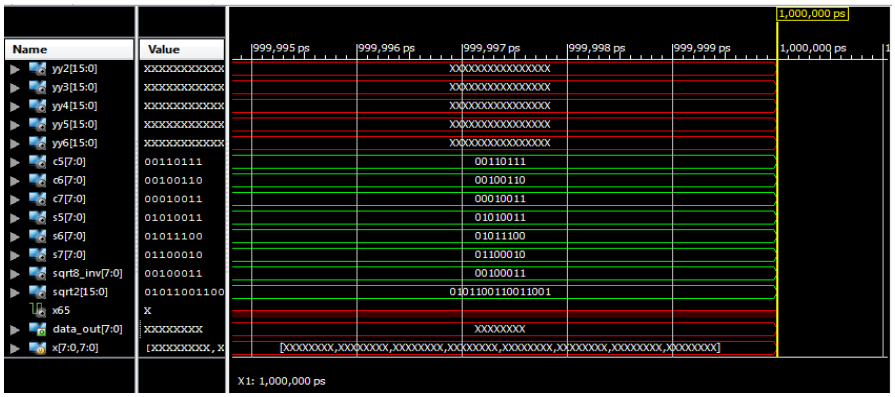

Fig.11. Schematic result of DCT unit

Table II. Comparison of power and delay of DCT and modified DCT

Parameter	Total power(mw)	Delay(ns)
DCT	22.76	8.053
Modified DCT	14.56	6.053

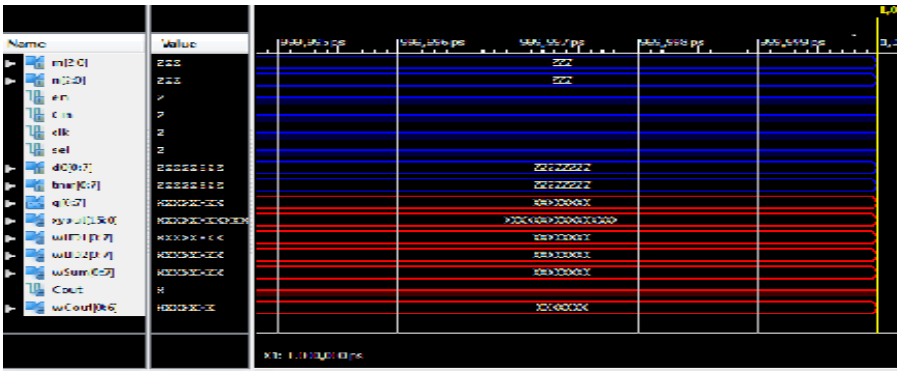

Fig.12. Schematic result of Frequency operation unit

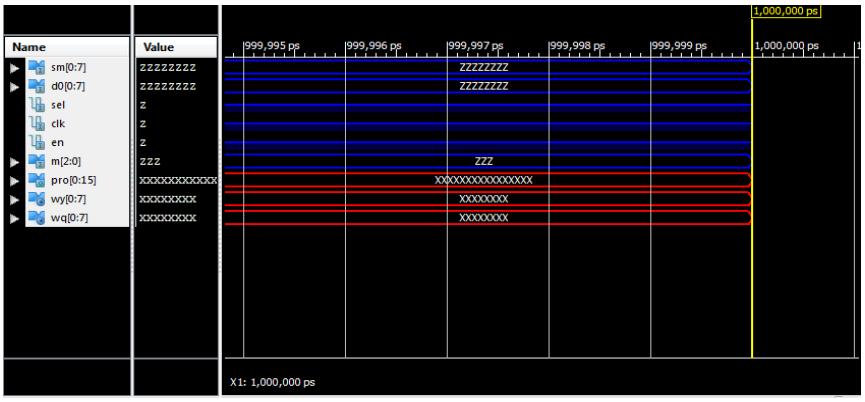

Fig.13. Schematic result of IDCT unit

The table shows the comparison of DCT and modified DCT using various parameters. It is noted that modified DCT using RCA has low power and delay.So modified DCT using ripple carry adder hass better power consumption and delay.

Table III. Device utilization ssummary of DCT and a modified DCT

Parameter	No: of slice LUT	No: of occupied slices	No: of bonded IOBs
DCT	2%	2%	35%
Modified DCT	1%	1%	15%

If we considered the area of modified DCT using RCA has lesser number of LUTS.

5 Conclusion

The architecture for phase unwrappping has been implemented using FPGA. The result has an advantage of low power consumption, dealy and area. Modified DCT based on RCA is more efficient. In case of area of modified DCT occupies more LUTs when compared to DCT. RCA is used to reduce the power, simple logic repeated adders and substractors are used. So thereby reduce the area viceverse reduce delay and power. When area reduced the power and time is also reduced.

References

1. Kim, M. K; Digital Holographic Microscopy, New York, NY, USA: Springer(2011)
2. Peter, A; Ivan, F.: Simple and accurate unwrapping phase of MR data. Measurement, vol. 42, pp. 737-741 (2009)
3. Hwang, W. J., Cheng, S. C.,Cheng, C. J.:Efficient phase unwrapping architecture for digital holographic microscopy. Sensors, vol. 11, pp. 9160–9181 (2011)
4. Bhaduri., et al.: Diffraction phase microscopy: Principles and applications in materials and life sciences. Adv. Opt. Photon., vol. 6, pp. 57–119 (2014)
5. Guo, X; Chen, Zhang, T.: Robust phase unwrapping algorithm based on least squares vol. 63, pp. 25-29 (2014)
6. Loffeld, O., Nies, H., Knedlik, S., Yu, W.: Phase unwrapping for SAR interferometry—A data fusion approach by Kalman filtering,.IEEE Trans.Geosci. Remote Sens., vol. 46, no. 1, pp. 47–58 (2008)
7. Mistry, P., Braganza, S., Kaeli, D., Leeser, M.: Accelerating phase unwrapping and affine transformations for optical quadrature microscopy using CUDA. in Proceedings of the Second Workshop on General Purpose Processing on GraphicsProcessing Units pp. 28–37 ACM (2009)
8. Rivera, M., Hernandez-Lopez, F.J., Gonzalez, A.: Phase unwrapping by accumulation of residual maps. Opt. Lasers Eng., vol. 64, pp. 51–58(2015)
9. Huang, M.J., He, Z. N.: Phase unwrapping through region reference algorithm and window-patching method. Opt. Commun. 203, 225–241 (2002)
10. Schnars, U., Jueptner, W.P.: Digital Holography, Springer-Verlag (2005)
11. Chavez, S., Xiang, Q. S., An, L.: Understanding maps in MRI: a new cutline phase unwrapping method. IEEE Trans. Med. Imaging 21, 966–977 (2002)
12. Bian. Y; Mercer. B: Weighted regularized preconditioned conjugate gradient (PCG) phase unwrapping method (2009)
13. Li, Z., Bao, Z., Suo, Z.: A joint image coregistration, phase noise suppression and phase unwrapping method based on subspace projection for multibaseline InSARSystems. IEEE Trans. Geosci. Remote Sens. 45, 584–591 (2007)
14. Ghiglia, D. C., Pritt, M. D.: Two-Dimensional Phase Unwrapping: Theory, Algorithm and Software. Hoboken, NJ, USA: Wiley (1998)