Synthesis and Selection of DCT Algorithms using Behavioral Synthesis-based Algorithm Space Exploration

Miodrag Potkonjak[‡] and Anantha Chandrakasan[†]
[‡]C&C Research Laboratories, NEC USA, Princeton, NJ
[†]Department of EECS, MIT, Cambridge, MA

ABSTRACT: Numerous fast algorithms for the Discrete Cosine Transform (DCT) have been proposed in image and video processing literature. Until recently, it has been difficult to compare different DCT algorithms and select one which is best suited for implementation under a given set of design goals and constraints. In this paper, we propose an approach for design space exploration at the algorithm and behavioral levels using high level synthesis tools. In particular, we study and compare the following nine DCT algorithms: Lee's, Wang's, DIT, DFT, QR, Givens, Arai, MCM, and direct algorithm. The main conclusion of this study is that the best choice among fast DCT algorithms depends on a particular set of design goals and constraints. Another important conclusion is that for almost all sets of implementation goals and constraints more than an order of magnitude improvement can be achieved using algorithm and behavioral design space exploration.

1.0 Motivation

The Discrete Cosine Transform (DCT) was introduced by Ahmed, Natarjan, and Rao in 1974 [Ahm74]. It has found a wide spectrum of applications in image and video processing and several other signal processing application domains [Rao90] mainly due to its remarkable similarity to the statistically optimal Karhunen-Loeve transform when highly correlated signals are processed. Another attractive feature of DCT is its relatively low implementation complexity. Recently DCT has been promoted to a position of ubiquitous computation due to its acceptance as a part several popular and widely used image and video compression standards (H261, JPEG, MPEG) [Pir95]. In the last 20 years numerous fast algorithms and several VLSI custom integrated circuits (IC) implementations have been proposed for optimizing the application specific IC (ASIC) implementation cost [Wan85, Sue86, Yip84, Yip85, Ara88, Vet86, Loe88, Rao90, Pot94a, Pir95].

However, the comparison of several fast DCT algorithms and in particular their IC implementations for a given set of design goals and constraints is often very involved, time consuming, and cumbersome [Rao90]. Choosing the "best" fast DCT algorithm for a particular image or video application is a non-trivial task. In such situations, behavioral synthesis tools provide an excellent option for rapid exploration of the algorithmic design space [Rab91, McF90]. After a long period of academic research, behavioral synthesis recently entered into a more mature phase where several research and commercial behavioral synthesis tools have become available, providing a reliable and fast path from functional specification to custom ASIC implementation.

In this paper we demonstrate how algorithm space exploration can be rapidly conducted using high level synthesis and silicon compilation tools. We evaluate nine different 2-dimensional 8 X 8 DCT algorithms using a set of behavioral synthesis tools. The study provides insights and supporting experimental data which clearly indicate the importance of algorithm space exploration for video algorithms. Furthermore for the first time we present data which compare different DCT algorithms with respect to increasingly important design metrics such as power consumption and latency.

2.0 Evaluation Strategy and Experimental Set-Up

For synthesis and algorithm space exploration we used the high level synthesis system, HYPER [Rab91] and the silicon compiler, LAGER [Bro92] from the University of California at Berkeley. We selected the HYPER high level synthesis tool because it provides exceptionally strong support for design space exploration. We coded all DCT algorithms in the applicative, functional, DSP programming language SILAGE which HYPER translates into the CDFG format. After the initial functional simulation, HYPER synthesizes the targeted design under the user specified set of timing (throughput) constraints [Rab91]. The synthesis procedure includes several high

0-8186-7310-9/95 $4.00

algorithm	# of additions	# of multiplications	# additions/ subtractions (T)	# of shifts (T)
direct	56	64	218	198
DIF	20	12	62	47
DIT	35	14	72	49
wang	26	21	95	81
lee	20	18	54	46
QR	25	16	79	60
givens	20	28	111	106
arai	29	5	47	20
mcm	64	30	104	78

Table 1: Comparison of Fast algorithms for DCT: the number of operations before and after conversion of constant multiplication to shifts and additions (T) .

algorithm	T = 1	T = 2	T = 4	T = 5	T = 10	T = 20	T = 50
direct	1668.33	843.06	350.58	274.45	102.36	68.10	31.45
DIF	138.96	73.04	49.98	42.04	10.07	8.99	8.36
DIT	270.50	139.36	88.39	77.51	17.77	10.66	9.82
wang	315.01	154.23	89.84	77.90	21.97	15.25	10.07
lee	244.24	121.16	73.55	63.91	21.48	10.03	9.38
QR	240.89	118.74	66.52	62.25	22.27	10.62	9.26
givens	312.56	174.02	24.65	64.47	26.03	14.34	9.43
arai	125.69	72.22	48.83	41.43	11.10	9.41	8.58
mcm	689.22	322.15	179.61	150.95	46.14	11.31	9.49

Table 2: Comparison of Fast algorithms for DCT: area of implementations under several different throughput requirements.

algorithm	T = 1	T = 2	T = 4	T = 5	T = 10	T = 20	T = 50
direct	377.87	209.60	105.21	87.30	55.89	52.48	51.42
DIF	19.98	15.36	14.77	14.02	10.49	10.66	11.25
DIT	32.44	26.87	24.07	23.43	14.38	13.63	14.56
wang	36.13	29.54	25.95	25.27	16.48	16.09	16.51
lee	27.36	22.82	20.68	20.05	13.89	12.90	13.97
QR	32.87	20.01	26.22	26.55	19.26	18.51	17.70
givens	40.25	34.36	31.35	31.24	25.57	24.65	25.69
arai	17.73	15.15	14.99	14.96	9.04	9.07	10.08
mcm	84.77	66.32	56.44	53.95	33.94	29.13	29.56

Table 3: .Comparison of Fast algorithms for DCT: power (nJ/sample) of implementations under several different throughput requirements.

algorithm	T = 1	T = 2	T = 4	T = 5	T = 10	T = 20	T = 50
direct	1166.69	325.59	228.35	194.49	34.95	20.98	7.24
DIF	184.98	46.22	33.08	31.22	6.91	5.02	4.30
DIT	196.98	51.17	37.72	34.43	6.98	5.17	4.51
wang	223.88	53.39	36.64	33.86	6.87	6.09	4.82
lee	187.77	50.09	31.52	28.65	6.58	5.76	4.47
QR	188.82	46.87	31.22	27.77	5.59	4.04	3.82
givens	249.01	57.79	45.61	39.95	7.77	5.97	4.78
arai	184.00	45.33	31.57	30.14	6.69	4.22	3.75
mcm	192.67	45.99	38.99	36.77	12.44	6.84	4.80

Table 4: Comparison of Fast algorithms for DCT: area of implementations under several different throughput requirements after substitution of constant multiplications with shifts and additions.

level synthesis steps, such as resource allocation, scheduling, assignment, hardware selection and hardware mapping [Rab91]. During the synthesis process a number of behavioral transformations can be optionally invoked to improve one or more design characteristics of designs [Pot94b]. In particular, as it shown later, we used pipelining and constant multiplication to shift and additions conversion to significantly reduce implementation area for all algorithms.

We synthesized, under the identical throughput constraints, the following nine DCT algorithms: Lee - Lee's recursive sparse matrix factorization algorithm [Lee84], Wang- Suehiro-Hatori's version of the Wang planar rotation-based sparse matrix factorization DCT [Wan85, Sue86], DIT - recursive decimation in time algorithm [Yip84], DIF - recursive decimation in frequency algorithm [Yip85], QR - QR decomposition based hybrid planar rotation algorithm [Vet86], givens - Givens rotation-based algorithm [Loe88], Arai - Arai-Agui-Nakajima SQ algorithm [Ara88], MCM - automatically synthesized algorithm, which applies only one multiple constant multiplication transformation on the generic, direct, DCT transform [Pot94a], and direct- the direct, generic definition of DCT algorithm. All DCT transforms are evaluated as two dimensional 8 X 8 transforms, assuming that two dimensional transformations are formed using 16 1D DCTs and appropriate transposition of data [Rao90]. All designs are studied assuming 12-bit coefficients and 9 bit data. This assumption can be easily altered to any other precision criteria, including one based on simulation results.

3.0 Experiments: Results and Analysis

Table 1 shows the number of operations in nine DCT algorithms before and after the application of substitution of constant multiplication by shifts and additions. It is apparent that the Arai DCT fast algorithm requires the smallest number of multiplication and the smallest total number of operations both before and after the application of transformations. The substitution of multiplications with constants by shifts and additions/substraction is conducted using the canonical signed digit algorithm [Rab91].

Tables 2 and 3 show the ASIC area (in mm^2) and power consumption (nJ/sample) of the designs. On four fastest implementation of each DCT algorithm the maximal pipelining was applied in order to satisfy strict throughput requirements. Table 4 shows area in mm^2 of the designs after the application of substitution of constant multiplication by shifts and additions. It is interesting to note that at the lowest throughput requirements almost all fast DCT algorithms require similar area. The smallest design is DIF which has 7 more multiplications than the Arai DCT. This is mainly due to reduced interconnect and register requirements. However, after the application of transformations, the smallest design is one which corresponds to the Arai DCT. Furthermore, note that the largest transformed design (direct) has smaller area than the smallest design implemented without the application of transformations (DIF).

algorithm	Latency [nsec]
direct	380
DIF	600
DIT	620
wang	600
lee	560
QR	560
givens	600
Arai	560
mcm	340

Table 5: The latency of different DCT algorithms after the application of substitution of constant multiplications with shifts and additions/subtractions

Table 5 shows the latencies of the nine DCT algorithms. Latency is the minimal time interval required to obtain all outputs, from the time when all input data is available. Latency is important design metrics when DCT is used as a part of feedback loop, as it is the case in several video and image standards. During the calculation of latency, it is assumed that no resource constraints are imposed. Interestingly, two designs with the smallest latencies are the direct and mcm DCTs which have the largest number of operations.

We conducted several other measurements, but due to lack of space they are not included in this version of the paper. In all tables the integer multiplies of the identical throughput constraint, T, equal to 576 nsec are used.

All results were generated and analyzed in a time span of six hours, which demonstrates that efficiency of the HYPER high level synthesis system is high enough to properly address the design space exploration. A number

of conclusions can be drawn from the experimental data, including:

(1) Transformations often have significantly higher impact on both area and power than just the computational differences between different fast algorithms. However, their application should be properly undertaken due to the potential side-effects of transformations. For example, at very high throughputs, due to a need for excessive pipelining, the implementation areas increased after substitution of constant multiplications with shifts and additions.

(2) Arai DCT algorithms is often, due to its low computational requirements, one of the best choices. However, several DCT algorithms which do not scale coefficients are competitive for several optimization scenarios. The best algorithm for a particular set of design goals and constraints can be selected only after complete synthesis. In majority of designs the area was dominated by interconnect and registers.

(3) Importance of simultaneous consideration of algorithms selection and behavioral transformations is very high. For example, when transformed QR algorithms is used instead of direct DCT under the throughput requirements of 10T, the difference in final implementation area is greater by a factor of 18 (102.36 mm^2 vs. 5.59 mm^2). Similarly, even when transformations are not used, the difference in power consumption between the direct DCT algorithm and the Arai fast DCT algorithms is more than a factor of 21 (377.87 nJ/sample vs. 17.73 nJ/sample) for the same throughput requirement of 576 nsec.

4.0 Conclusion

We introduced an approach for fast and accurate algorithm and behavioral synthesis-level design solution space exploration using high level synthesis techniques and tools. In particular, we concurrently studied nine different 2D 8 X 8 DCT algorithms (Lee's, Wang's, DIT, DIF, Arai's, QR, Givens, mcm, and direct - generic) and their suitability for custom ASIC implementation under several scenarios of design goals and constraints. It has been shown that the approach offers more than an order of magnitude improvements in several design metrics, such as area and power, using proper algorithm and behavioral synthesis options.

5.0 References

[Ahm74] N. Ahmed, T. Natarajan, K.R. Rao: "Discrete cosine transform", IEEE Transactions on Communications, Vol. 23, No. 1, pp. 90-93, 1974.

[Ara88] Y. Arai, T. Agui, M. Nakadjima, "A Fast DCT-SQ scheme for images", Trans. IEICE, Vol. E71, No. 11, pp. 1095-1097, 1988.

[Bro92] R. W. Brodersen, editor. *Anatomy of a Silicon Compiler.* Kluwer Academic Publishers, 1992.

[Lee84] B.G. Lee: "A new algorithm to compute the discrete cosine transform", IEEE Transactions on Acoustic, Signal, and Speech Processing, Vol. 32, No. 12, pp. 1243-1245, 1984.

[Loe88] C. Loeffler, A. Ligtenberg, G.S. Moschyz: "Algorithm-architecture mapping for custom DSP chips", International Symposium on Circuits and Systems, pp. 1953-1956, 1988.

[McF90] M.C. McFarland, A.C. Parker, R. Camposano: "The High Level Synthesis of Digital Systems", Proc. of the IEEE, Vol. 78, No. 2, pp. 301-317, 1990.

[Pen93] W.B. Pennebaker, J.L. Mitchell: "JPEG still image data compression standard", Van Nostrand, New York, NY, 1993.

[Pot91] M. Potkonjak, J. Rabaey: "Optimizing the Resource Utilization Using Transformations", Proc. IEEE ICCAD-91, Santa Clara, pp. 88-91, November 1991.

[Pot94a] M. Potkonjak, M.B. Srivastava, A. Chandrakasan, "Efficient Substitution of Multiple Constant Multiplications by Shifts and Additions using Iterative PairwiseMatching", 31th ACM/IEEE DAC Design Automation Conference, pp. 189-194,June 1994.

[Pot94b] M. Potkonjak, J. Rabaey, "Optimizing Throughput and Resource Utilization using Pipelining: Transformation Based Approach", Journal of VLSI Signal Processing,Vol. 8, No. 2, pp. 117-130, October 1994.

[Rab91] J. Rabaey, C. Chu, P. Hoang, M. Potkonjak: "Fast Prototyping of Datapath-Intensive Architectures", IEEE Design and Test of Computers, Vol. 8, No. 2, pp. 40-51, June 1991.

[Rab94] J.M. Rabaey, M. Potkonjak: "Estimating Implementation Bounds for Real Time DSP Application Specific Circuits", IEEE Trans. on CAD, Vol. 13, No. 6, to be published, 1994.

[Rao90] K.R. Rao, P. Yip: "Discrete Cosine Transform", Academic Press, Inc., San Diego, CA 1990.

[Sue86] N. Suehiro, M. Hatori: "Fast Algorithms for the DFT and other sinusoidal transforms", IEEE Transactions on Acoustic, Signal, and Speech Processing, Vol. 34, No. 6, pp. 642-644, 1986.

[Vet86] M. Vetterli, A. Ligtenberg: "A discrete Fourier-cosine transform chip", IEEE Journal on Selected Areas in Communications, Vol. 4, No. 1, pp. 49-61, 1986.

[Wan85] Z. Wang: "Fast Algorithms for discrete W transform and for the discrete Fourier Transform", IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. 32, No. 8, pp.803-816, 1994.

[Yip84] P. Yip, K.R. Rao: "Fast DIT algorithms for DCT and DST", Circuits, Systems, and Signal Processing, Vol. 3, No. 4, pp.387-408, 1984.

[Yip85] P. Yip, K.R. Rao: "DIF algorithms for DCT and DST", International Conference on Acoustic, Signal, and Speech Processing, pp. 776-779, 1985.