Performance-Cost Analyses Software for H.264 Forward/Inverse Integer Transform

Trang T.T. Do, Thinh M. Le, Binh P. Nguyen, Yajun Ha
Department of ECE, National University of Singapore, Singapore
dothutrang@nus.edu.sg

Abstract

In the literature, Data Throughput rate per Unit Area (DTUA) has been used as the sole metric to evaluate the effectiveness of H.264 Forward/Inverse Integer Transform (FIT/IIT) designs. However, other than throughput and circuit area involved in DTUA, interconnection, power and delay are not considered. In this paper, we first summarize the Performance-Cost Metric (PCM) technique, where PCM is defined as the ratio of data throughput over the design cost including power, area, delay, and issues associated with interconnections in sub-micron designs. Compared to DTUA, PCM facilitates more comprehensive comparisons for VLSI designs, including FIT/IIT. The contribution of this paper is the software that helps facilitate the use of the PCM technique. When using this software, users are asked to enter some preliminary parameters of their design. Based on the given parameters and the reference designs managed using the software, it then analyzes and provides the possible boundaries of the users' design in order to have better PCMs compared to the reference designs. In addition, it can also export comparison results among different designs. The software is flexibly designed in order to facilitate the use of not only our PCM technique in FIT/IIT designs, but also different metrics in other architectures.

1 Introduction

H.264 [1]-[2] video compression standard provides better video quality at substantially lower bit-rate than the previous standards by the adoption of a number of new coding tools. One of the coding tools, Integer Transform (IT), is based on the Discrete Cosine Transform of the previous standards, e.g. MPEG-4 [3], with the avoidance of mismatch problem and the reduction of computational complexity. Four transforms are used in total: a Hadamard transform for the 4×4 array of luma DC coefficients in intra macroblocks predicted in 16×16 mode, a Hadamard transform for the 2×2 array of chroma DC coefficients in every macroblock, and 4×4 and 8×8 DCT-based transforms for all residual blocks in fidelity range extensions [4].

Several Forward IT (FIT) and Inverse IT (IIT) designs have been reported in the literature [5]-[11]. In general, the primary objective of all the existing designs is to achieve high performance, whereas the secondary objective is to reduce design costs. High performance is often in terms of high throughput with preferably high operating frequency [5]-[8], [10]-[11]. Low cost is seen as having low power [6] or low gate count [5], [8]. In the literature, throughput is either computed as the maximum amount of data input at a time [7], [9] or as the average amount of data processed in one time unit [5], [6], [8], [10]-[11]. Typically, the former technique produces a higher throughput, while the latter is used in practice.

Data Throughput rate per Unit Area ($DTUA$) was first reported in the literature for Discrete Fourier Transform design. $DTUA$ is used to compare FIT/IIT designs in [5], [8]. Other than $DTUA$, comparisons using throughput, gate count, or power among designs have been arbitrary. Evaluation of designs using only throughput and circuit area is insufficient. Thus, a metric - comprising most if not all performance and cost parameters - is needed.

In this paper, the Performance-Cost Metric (PCM) technique facilitating comparisons among designs is summarized, which is proposed in our other manuscript. Based on PCM, a Performance-Cost Analysis Software (PCAS) is developed, which can analyze design boundaries and perform rapid comparison among different designs. The rest of the paper is organized as follows: In Section 2, FIT/IIT design costs are analyzed, followed by the definition and description of performance-cost metric in Section 3. In Sections 4, a detailed discussion of PCM in comparison with $DTUA$ through different designs in literature is presented. PCAS is introduced in Section 5. The paper ends with conclusions in Section 6.

2 Cost Analyses for FIT/IIT Designs

2.1 Estimation of Power Consumption

2.1.1 Background

In CMOS technology, power consumption can be approximated using dynamic component

$$P_{dyn} = \alpha C_L V_{DD}^2 f \tag{1}$$

where α, C_L, V_{DD}, and f represent switching activity, load capacitance, supply voltage, and operating frequency, respectively. Dynamic power is proportional to the load capacitance C_L. Since it is shown that gate and interconnect capacitances contribute significantly to the total load capacitance [12], total load capacitance can be approximated as the sum of gate and interconnect capacitances:

$$C_L = C_G + C_{interconnect} \tag{2}$$

2.1.2 Gate Capacitance

The total gate capacitance is proportional to the gate count of the design. Using the 2-input NAND gate (NAND2) in the standard cell library of a particular technology as a unit, C_G can be defined as:

$$C_G = \beta G \tag{3}$$

where β and G are the gate capacitance and gate count of a unit NAND2. According to [13]:

$$\beta = C_{G\_NMOS} + C_{G\_PMOS} = C_{ox} L_{channel} (W_{NMOS} + W_{PMOS}) \tag{4}$$

where C_{ox}, $L_{channel}$, W_{NMOS} and W_{PMOS} are gate oxide capacitance per unit area, channel length, NMOS and PMOS channel width inside the unit gate, respectively.

2.1.3 Interconnect Capacitance

Interconnect capacitance, on the other hand, includes three components: i) *parallel-plate capacitance*; ii) *fringe capacitance* between the side walls of the wires and substrate; and iii) *inter-wire capacitance* [13]

$$C_{interconnect} = C_{pp} + C_{fringe} + C_{interwire} \tag{5}$$

$C_{interwire}$ dominates the total interconnect capacitance under submicron technology [14]. $C_{interwire}$ between two conservative wires is expressed as follows:

$$C_{interwire\_2} = (\frac{\varepsilon_{di}}{t_{di}})HL \tag{6}$$

where ε_{di} and t_{di} are permittivity and thickness of the dielectric layer, and H and L are length and thickness of the interconnect. Assume that a layer contains N parallel wires, its inter-wire capacitance is computed in (7). When N is large, (7) becomes (8).

$$C_{interwire\_N} = (N-1)(\frac{\varepsilon_{di}}{t_{di}})HL \tag{7}$$

$$C_{interwire\_N} \approx N(\frac{\varepsilon_{di}}{t_{di}})HL \tag{8}$$

2.1.4 Inter-wire Capacitance

An IC has several layers of wires generally classified as local and global. In order to estimate the inter-wire capacitance using (8), the number of wires N for each layer is estimated. Using Rent's Rule [15], the number of nets including module terminals for a module containing B blocks, where each block has an average of K pins (terminals), N, can be estimated as:

$$N = K(B + B^r)/2 \tag{9}$$

where r is a constant depending on block type. Now we apply (9) to find the number of nets at each layer for an IC. Local wire layers are used to connect gates, hence gates now can be considered as blocks in (9). Assume that the IC contains G NAND2 gate. When blocks are gate array, $r = 0.5$ [15], the total number of local wires connecting G gates is:

$$N_{LC} = 3(G + G^{0.5})/2 = 1.5(G + \sqrt{G}) \tag{10}$$

Global wire layers are used to connect IP blocks in a SoC, hence IP blocks now can be considered as blocks in (9). Assume that each IP block has K pins (terminals or input/output), the total number of wires in global layers is:

$$N_{GB} = K(B + B^r)/2 \tag{11}$$

Also, assume that all global wires have the same height and length, and all local wires have the same height and length, inter-wire capacitance is:

$$C_{interwire} = N_{LC}(\frac{\varepsilon_{di}}{t_{di}})H_{LC}L_{LC} + N_{GB}(\frac{\varepsilon_{di}}{t_{di}})H_{GB}L_{GB} \tag{12}$$

$$C_{interwire} = 1.5(\frac{\varepsilon_{di}}{t_{di}})H_{LC}L_{LC}(G + \sqrt{G}) + \frac{1}{2}(B + B^r)(\frac{\varepsilon_{di}}{t_{di}})H_{GB}L_{GB}K \tag{13}$$

Let us define:

$$\gamma = 1.5(\frac{\varepsilon_{di}}{t_{di}})H_{LC}L_{LC} \tag{14}$$

$$\theta = \frac{1}{2}(B + B^r)(\frac{\varepsilon_{di}}{t_{di}})H_{GB}L_{GB} \tag{15}$$

then (13) becomes:

$$C_{interwire} = \gamma(G + \sqrt{G}) + \theta K \tag{16}$$

2.1.5 Formula for Power Consumption

Based on (1-3) and (16), the total power consumption of an IC can be estimated as follow:

$$P = \alpha[\beta G + \gamma(G + \sqrt{G}) + \theta K]V_{DD}^2 f \tag{17}$$

2.2 Estimation of Circuit Area

The IC area can be estimated as the maximum among the followings: i) area of total number of logic gates, ii) area of local interconnect layers, and iii) area of global interconnect layers.

$$A = \max\{A_G, A_{LC}, A_{GB}\} \tag{18}$$

However, in today's submicron era, transistors are becoming relatively small, and chips are mostly made of wires [16]. As stated in [17], interconnects have dominant impact on IC area. As the number of wires can be estimated using (10-11), assuming they need number of layers, η_{LC} and η_{GB}, to build all local and global wires, respectively. If W, S, L are the width, spacing, and length of wires, area of local and global layers can be estimated as:

$$A_{LC} \approx 1.5(G + \sqrt{G})(W_{LC} + S_{LC})L_{LC}/\eta_{LC} \tag{19}$$

$$A_{GB} \approx \frac{1}{2}W(B + B^r)(W_{GB} + S_{GB})L_{GB}/\eta_{GB} \tag{20}$$

Let us define:

$$\chi = 1.5(W_{LC} + S_{LC})L_{LC} / \eta_{LC} \tag{21}$$

$$\xi = \frac{1}{2}(B + B^r)(W_{GB} + S_{GB})L_{GB} / \eta_{GB} \tag{22}$$

thus the area of the layers and the total IC area can be estimated as:

$$A_{LC} = \chi(G + \sqrt{G}) \quad (23) \qquad A_{GB} = \xi K \quad (24)$$

$$A = \max\{A_{LC}, A_{GB}\} = \max\{\chi(G + \sqrt{G}), \xi K\} \tag{25}$$

Since the size of the global wire is much larger than that of the local one, when the average number of pins of IP blocks, K, is large enough, A_{GB} will dominate, or IC area depends on K. On the other hand, if K is small, IC area can be approximated by A_{LC}, and IC area depends on G.

2.3 Estimation of Design Costs

Design cost is defined as the product of power, area, and total delay $Cost = P \times A \times D_s$ (in Wm^2s) (26)

where P is in W, A in m^2, and D_s in seconds. The total delay D of a processing module includes I/O delays and processing delay to complete a given task. Depending on the emphasis, P, A or D_s may carry different exponents. In this case, we assume they are equally important and thus assign exponent of 1 to each. From (17) and (25):

$$Cost = \alpha[\beta G + \gamma(G + \sqrt{G}) + \theta K]V_{DD}^2 f \times \max\{\chi(G + \sqrt{G}), \xi K\} \times D_s \tag{27}$$

As G is normally larger than 10,000, $\sqrt{G}$ is smaller than 1% of G. Thus, $\sqrt{G}$ can be neglected and (27) becomes:

$$Cost \approx \alpha[(\beta + \gamma)G + \theta K]V_{DD}^2 f \times \max\{\chi G, \xi K\} \times D_s \tag{28}$$

Sylvester and Keutzer [18] showed that gates and local interconnects power consumption dominate total power consumption. In (28), power consumption due to global interconnects becomes $\alpha\theta K V_{DD}^2 f$, while power due to gates and local interconnects becomes $\alpha(\beta + \gamma)GV_{DD}^2 f$. Hence,

$$Cost \approx \alpha(\beta + \gamma)GV_{DD}^2 f \times \max\{\chi G, \xi K\} \times D_s \tag{29}$$

Since in the standard 2-input NAND cells, PMOS width is normally equal to about 0.8 times of NMOS width [19], while NMOS width can be approximately scaled with technology (channel length) at about 3.5 times [20]. Therefore, (4) becomes:

$$\beta \approx 1.8 C_{ox} L_{channel} W_{NMOS} \approx 6.3 C_{ox} L_{channel}^2 \tag{30}$$

In addition, the gate oxide capacitance per unit area C_{ox} is calculated as follows [13]:

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} = \frac{\varepsilon_{rox}\varepsilon_0}{t_{ox}} \quad (31) \qquad \varepsilon_0 = 8.854 \times 10^{-12} F/m \quad (32)$$

where ε_{ox}, ε_0, ε_{rox}, t_{ox} are permittivity of SiO$_2$, permittivity of free space (electric constant or vacuum permittivity), relative static permittivity (or dielectric constant) of SiO$_2$, and gate oxide thickness, respectively. Thus, (30) becomes:

$$\beta \approx 1.8 C_{ox} L_{channel} W_{NMOS} \approx 6.3 \frac{\varepsilon_{rox}\varepsilon_0}{t_{ox}} L_{channel}^2 \tag{33}$$

In a given SoC-based IC, if the average number of pins per IP block is small, $\chi G > \xi K$, (29) becomes:

$$Cost_G \approx (\psi G V_{DD}^2 f).G.D_s \tag{34}$$

where $\psi = \alpha(\beta + \gamma)\chi$ (35)

$$\psi = 1.5\alpha\left(6.3\frac{\varepsilon_{rox}\varepsilon_0}{t_{ox}} L_{channel}^2 + 1.5\frac{\varepsilon_{di}}{t_{di}} H_{LC} L_{LC}\right)(W_{LC} + S_{LC})\frac{L_{LC}}{\eta_{LC}} \tag{36}$$

$$\psi = 1.5\alpha\varepsilon_0\left(6.3\frac{\varepsilon_{rox}}{t_{ox}} L_{channel}^2 + 1.5\frac{\varepsilon_{rdi}}{t_{di}} H_{LC} L_{LC}\right)(W_{LC} + S_{LC})\frac{L_{LC}}{\eta_{LC}} \tag{37}$$

where ε_{rdi} is relative static permittivity of the dielectric layer under the local wires.

On the other hand, if the average number of pins per IP block is large, $\chi G < \xi K$, (29) becomes:

$$Cost_K \approx (\varphi G V_{DD}^2 f).K.D_s \tag{38}$$

where $\varphi = \alpha(\beta + \gamma)\xi$ (39)

$$\varphi = \frac{1}{2}\alpha\varepsilon_0\left(6.3\frac{\varepsilon_{rox}}{t_{ox}} L_{channel}^2 + 1.5\frac{\varepsilon_{rdi}}{t_{di}} H_{LC} L_{LC}\right)(B + B^r)(W_{GB} + S_{GB})\frac{L_{GB}}{\eta_{GB}} \tag{40}$$

In summary, given a system with IP blocks, design cost metric depends on whether the average number of pins per IP block is small or large. If it is small, the area due to local interconnect dominates and $Cost_G$ (34) can be used. If it is exceedingly large, $Cost_K$ (38) can be used as the global dominates. In both conditions, cost is proportional to delay.

In both cost formulas, ψ and φ depend on technology design rules and process. In particular, they are proportional to total number of gates and local interconnect capacitance, which depends on gate length $L_{channel}$, gate oxide thickness t_{ox}, and local interconnect thickness H_{LC} and length L_{LC}.

3 Summary of Performance-Cost Metric for FIT/IIT Designs

Throughput of a SoC-based data processing design is commonly measured in the number of pixel-per-second (pps). Throughput of FIT/IIT modules for $n \times n$ blocks can be computed by total delay (second):

$$T_{pps} = n^2 / D_s \tag{41}$$

In order to compare among designs with different performance and costs, we define a *performance-cost metric* (*PCM)* as follows:

$$PCM = \frac{Performance}{Cost} = \frac{T_{pps}}{Cost} \quad (\#\text{ of } pixels/Wm^2s^2) \tag{42}$$

TABLE I. PERFORMANCE-COST FUNCTION PCM COMPARISON OF PRE-LAYOUT SYNTHESIZED DESIGNS

Design	**FIT/ IIT**	**Type of transform**	**Quan-tiza-tion**	**Tech.** (µm)	**Gate count** (K)	**Speed** (MHz)	**Bus width** (pels/bits)	**Delay** (cycles)	**Throughput**$^{\wedge}$ (ppc/Mpps)	**C$_G$** (GV^{2}ppc)	**C$_K$** (MV^{2}bitppc)	**PCMG** (Hz/MV^{2}ppc^2)	**PCMK** (Hz/KV^{2}bitppc^2)	**DTUA** (pps)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
[5]	FIT	4x4(H),8x8$^{\#}$	Yes	0.35	115.3	80.0	128/2048	2	32.0/2560.0	289.5	5143	138.1	7.8	22.2
[6]	**IIT**	**4x4(H),8x8$^{\#}$**	**No**	**0.35**	**20.7**	**27.0**	**64/768$^{\&}$**	**5**	**12.8/345.6**	**23.3**	**865.6**	**231.4**	**6.2**	**16.7**
[7]	IIT	All	Yes	0.35	21.5	150.0	8/128	21	3.0/450.0	105.7	629.4	67.6	11.4	20.9
[5]	IIT	4x4(H),8x8$^{\#}$	Yes	0.35	103.9	82.0	128/2048	2	32.0/2624.0	235.1	4634.5	174.4	8.8	25.3
[5]	FIT	4x4(H),8x8$^{\#}$	Yes	0.18	162.1	77.0	128/2048	2	32.0/2464.0	170.3	2151.2	226.1	17.9	15.2
[8]	FIT	All	Yes	0.18	62.2	162.1	16/192	8	8.0/1296.8	100.3	309.5	202.1	65.4	20.8
[9]	**IIT**	**All**	**No**	**0.18**	**9.5**	**125.0**	**24/288$^{\&}$**	**19**	**3.4/425.0**	**5.6**	**168.4**	**1184.1**	**39.1**	**44.7**
[10]	**IIT**	**All**	**No**	**0.18**	**18.1**	**100.0**	**128/1920**	**16**	**4.0/400.0**	**17**	**1801.5**	**368**	**3.5**	**22.1**
[8]$^{(*)}$	**IIT**	**All**	**No**	**0.18**	**33.3**	**230.9**	**16/192**	**8**	**8.0/1847.2**	**28.7**	**165.7**	**1004.2**	**175.4**	**55.5**
[5]	IIT	4x4(H),8x8$^{\#}$	Yes	0.18	141.3	83.0	128/2048	2	32.0/2656.0	138.1	1937.6	300.4	21.4	18.8
[8]	IIT	All	Yes	0.18	47.3	230.9	16/192	8	8.0/1847.2	58	235.4	497.7	122.0	39.1

All: Design supports all IITs, including 4x4, 8x8 IIT, and 4x4, 2x2 Hadamard transforms; ^: Throughput is computed as the maximum amount of data input at a time; (*) The quantization in design in [8] is removed to facilitate comparison with [9] and [10]; #: Design supports all IITs, except 2x2 Hadamard inverse transforms; &: Assuming the bit-depth of 12 bits for one pixel as it was not reported.

Applying (42) for FIT/IIT module design which processes $n \times n$ blocks, we have:

$$PCM = \frac{n^2 / D_s}{Cost} \tag{43}$$

Since the design cost can be estimated using either (34) or (38) depending on the average number of pins per IP block, *PCM* now becomes:

$$PCM_G \approx \frac{n^2 / D_s}{(\psi G V_{DD}^2 f).G.D_s} \quad (44) \qquad PCM_K \approx \frac{n^2 / D_s}{(\varphi G V_{DD}^2 f).K.D_s} \quad (45)$$

In (44), gate count and delay are equally important and have the inverse square effect to PCM. In (45), delay has an inverse square effect, whereas gate count and average number of pins have an inverse linear effect on PCM.

In both formulas, PCM also depends on technology design rules and process through ψ and φ parameters. In order to facilitate comparisons among designs with the same block size *n*, the same number of blocks B, and to hide technology design rules and process parameters ψ or φ, we define *PCMG* and *PCMK* as follows:

$$PCMG = \frac{\psi PCM_G}{n^2} \approx \frac{1}{(G V_{DD}^2 f) G {D_s}^2} \tag{46}$$

$$PCMK = \frac{\varphi PCM_K}{n^2} \approx \frac{1}{(G V_{DD}^2 f) K {D_s}^2} \tag{47}$$

As the relationship between delays in second D_s and cycles D_c is $D_s = D_c t = D_c / f$ (48)

where t, f are the period (in seconds), and frequency, respectively, we have:

$$PCMG \approx \frac{1}{(G V_{DD}^2 f)\, G\, (\frac{D_c}{f})^2} \quad (49) \qquad PCMK \approx \frac{1}{(G V_{DD}^2 f)\, K\, (\frac{D_c}{f})^2} \quad (50)$$

Let us also define the technology-hidden C_G and C_K as:

$$C_G = \frac{Cost_G}{\psi} \approx (G V_{DD}^2 f).G.D_s = V_{DD}^2 G^2 D_c \tag{51}$$

$$C_K = \frac{Cost_K}{\varphi} \approx (G V_{DD}^2 f).K.D_s = V_{DD}^2 G K D_c \tag{52}$$

We also have:

$$PCMG \approx \frac{f}{V_{DD}^2 G^2 {D_c}^2} = \frac{f}{C_G D_c} \quad (53) \qquad PCMK \approx \frac{f}{V_{DD}^2 G K {D_c}^2} = \frac{f}{C_K D_c} \quad (54)$$

4 Discussion on PCMs on Different Designs and Metric Comparison to DTUA

The performance and cost of designs, [5]-[10] (Table I), will be evaluated based on the most complex transform 8×8. In order to facilitate the discussion, four groups of designs are classified, including 1) 0.35 µm FIT designs; 2) 0.35 µm IIT designs; 3) 0.18 µm FIT designs; and 4) 0.18 µm IIT designs. As group 1 has only one design, it will not be analyzed further.

An assumption is made for comparison among different designs that the SoC includes only FIT/IIT module. Therefore, designs using the same technology can have the same ψ or φ. As a result, C_G and C_K can be used for $Cost_G$ and $Cost_K$ comparison, and *PCMG* and *PCMK* for PCM_G and PCM_K comparison. Since the widths of the address bus and control bus are small compared to those of the data bus in FIT/IIT, the average number of pins K for FIT/IIT module can be approximated by the data bus width.

In group 2, design in [5] is 2.6 times higher in *PCMG* than that of [7]. On the other hand, design in [7] is 1.3 times higher in *PCMK* than that of [5]. Even though the *DTUA* and *PCMG* of [5] are relatively higher than those of [7], its *PCMK* is lower. The much larger bus width of [5] is reflected in the lower *PCMK* compared to that in [7]. The question is when to use *PCMG* and when to use *PCMK* ? It has shown that design in [5] has good performance due to its high throughput, high *DTUA*, and high *PCMG* value.

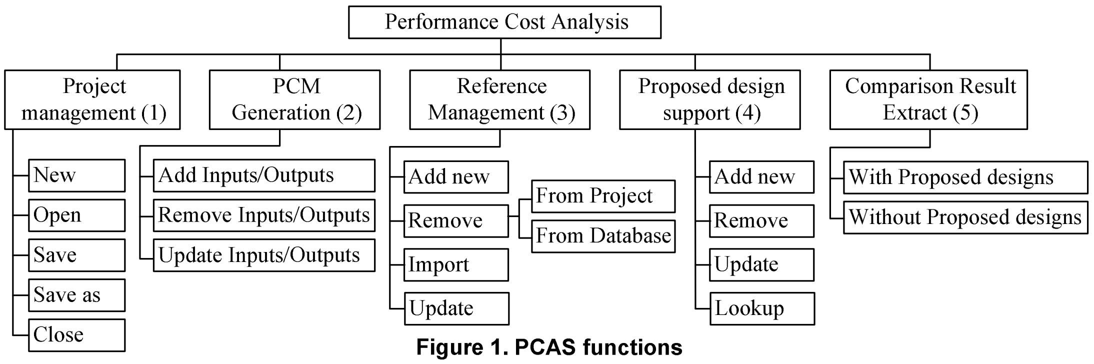


Figure 1. PCAS functions

However, when the technology shrinks, the interconnection issues start to dominate and thus $PCMK$ cost becomes large. It is suggested to use $PCMK$ in sub-micron technology or when bus width is exceedingly large, and to use $PCMG$ otherwise. In group 3, design in [8] has similar $PCMG$ and 3.7 times larger in $PCMK$ compared to [5]. Design in [8] has higher $DTUA$, comparable $PCMG$, and much higher $PCMK$ indicating that [8] is more favorable over [5]. In group 4, among designs without rescaling, [9] has the highest $PCMG$, which is 3.2 and 1.2 times higher than those of [10] and [8], respectively. Design in [9] specifically targets small gate count compared to others.

On the other hand, design in [8] has the highest $PCMK$, which is 4.4 and 49.6 times higher than those in [9] and [10]. Design in [8] is very much larger in $PCMK$ compared to that of [10]. This is because [10] requires extremely large I/O bus of 1920 bits, compared to 192 bits. This is another scenario where $DTUA$ clearly fails to report. Clearly, design in [10] has the lowest values in all metrics. [9] dominates in $PCMG$ (and $DTUA$), while [8] dominates in $PCMK$. Among designs with rescaling, [8] is 1.6 and 5.6 times higher in both $PCMG$ and $PCMK$. The three metrics show the same trend when $DTUA$, $PCMG$ and $PCMK$ of [8] is 2.1, 1.6 and 5.6 times larger than those of [5].

In general, $DTUA$ provides conventional view on assessing the performance-cost based on the area-only cost function. $PCMG$ provides the area-centric cost function, while $PCMK$ provides the interconnection-centric cost existed mostly in sub-micron designs with large number of pins. $DTUA$ has been used to assess the performance of a design using only throughput and area. If interconnections and large number of I/O pins and power consumption are concerned, $DTUA$ fails to report. Thus, the most performed designs should have highest values in $PCMG$ and $PCMK$.

5 Performance Cost Analysis Software

5.1 Overview of PCAS functions

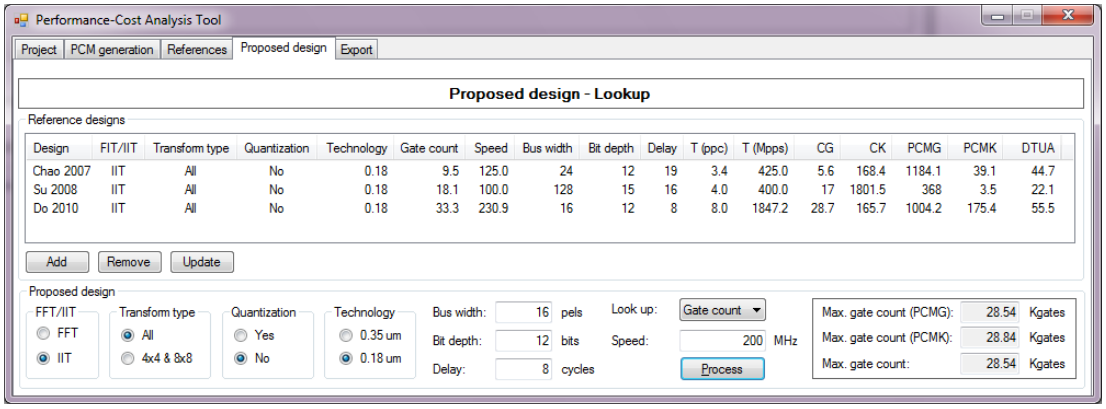
Based on PCM, a Performance-Cost Analysis Software (PCAS) has been developed to analyze and compare users' designs with reference designs. In particular, it helps to manage the references, generate different metric formulas, analyze the designs based on these metrics, lookup the allowed boundaries of their designs in order to have the best designs, and export comparison tables. In addition, due to a flexible function design, PCAS can be used not only for FIT/IITs using PCM but also other designs and metrics.

5.2 PCAS function description

The detail functions of PCAS are illustrated in Fig. 1. Branch 1 of the figure lists the functions for users to organize their work in projects. Functions that allow users to manage metrics, the central of analysis and comparison, are illustrated in Branch 2. In a project, different metrics, i.e. outputs, can be created and reused in other projects. Formulas of the metrics can be generated and modified based on variables, i.e. design input parameters. These inputs also can be added or removed. In the FIT/IIT case study, the outputs are throughput, C_G, C_K, $PCMG$, $PCMK$ and $DTUA$, while the inputs are G, f, K and D_C. There also are classified inputs such as whether FIT or IIT the module is, whether quantization step is included and which technology is used.

In addition, as published designs are also essential for analysis and comparison, PCAS provides functions to manage them for reference (Branch 3). Users can add or remove reference designs in current project or database. Once the references are added to a project, they are stored in the database and can be reused in other projects; thus, the users do not have to add them again. On the other hand, when a user removes a reference, it is removed by default only in the current project, not in database, since other projects might still use it. When a reference is added, the user needs to provide all the required input information of the reference. Besides adding and removing, importing function is available for users to import a reference from database to the current project. In addition, users can update information of a reference, leading to the update of the database and all the projects.

The main object of a project is the proposed design. Similar to reference designs, the proposed design is ma-



Proposed design - Lookup

Reference designs

Design	FIT/IIT	Transform type	Quantization	Technology	Gate count	Speed	Bus width	Bit depth	Delay	T (ppc)	T (Mpps)	CG	CK	PCMG	PCMK	DTUA
Chao 2007	IIT	All	No	0.18	9.5	125.0	24	12	19	3.4	425.0	5.6	168.4	1184.1	39.1	44.7
Su 2008	IIT	All	No	0.18	18.1	100.0	128	15	16	4.0	400.0	17	1801.5	368	3.5	22.1
Do 2010	IIT	All	No	0.18	33.3	230.9	16	12	8	8.0	1847.2	28.7	165.7	1004.2	175.4	55.5

Proposed design

FFT/IIT: FFT [✓ IIT]

Transform type: [✓ All] 4x4 & 8x8

Quantization: Yes [✓ No]

Technology: 0.35 um [✓ 0.18 um]

Bus width: 16 pels

Bit depth: 12 bits

Delay: 8 cycles

Look up: Gate count

Speed: 200 MHz

Max. gate count (PCMG): 28.54 Kgates

Max. gate count (PCMK): 28.84 Kgates

Max. gate count: 28.54 Kgates

Figure 2. Performance-Cost Analysis Software – Lookup function

naged by adding, removing and updating functions (Branch 4). In addition, when users design a new architecture, they probably need to know how good their design is compared to the existing ones. PCAS can provide a suggestion through the lookup function. The software allows users to choose one input and one output for the lookup function, in which the input is the lookup parameter for the new design and the output is the main metric. Next, the other input parameter values of the new design need to be provided. PCAS starts to compute all the metrics of the references, then search for the best designs based on the main metric, and finally compute the optimal value for the lookup parameter of the new design so that the users can use it as a goal to achieve a design which is better than the best reference designs. In the FIT/IIT case study, gate count or operating frequency (speed) can be chosen as the lookup parameter, while $PCMG$ and $PCMK$ can be selected as the main metrics. PCAS helps the users analyze the possible maximum gate count or minimum speed of their proposed designs based on its preliminary parameters in order to have a higher $PCMG$ and $PCMK$ compared to the highest $PCMG$ and $PCMK$ of the reference designs (Fig. 2).

Finally, PCAS can export comparison results to a table (Branch 5). The table looks similar to Table I and may or may not contain the information of the new design.

5.3 Optimal value calculation for look-up parameter in FIT/IIT case study.

In the FIT/IIT case study, assuming that gate count is chosen as the lookup parameter, and $PCMG$ is the main metric. Assuming that design X is the best among all reference designs, i.e. having the highest $PCMG$, and design A is the current design which is being processed by the lookup function. From (53), we have

$$PCMG_A \approx \frac{f_A}{V_{DD_A}^2 G_A^2 D_{C_A}{}^2} \quad (55) \qquad PCMG_X \approx \frac{f_B}{V_{DD_X}^2 G_X^2 D_{C_X}^2} \quad (56)$$

The current design A is better than the best reference design X when

$$PCMG_A > PCMG_X \quad (57) \qquad \frac{f_A}{V_{DD_A}^2 G_A^2 D_{C_A}{}^2} > \frac{f_X}{V_{DD_X}^2 G_X^2 D_{C_X}^2} \quad (58)$$

As technology is used to classify the designs, we only select the same technology for comparison. This means A and X have the same working voltage V_{DD}. So we have:

$$G_A < \frac{G_X D_{C_X}}{D_{C_A}} \sqrt{\frac{f_A}{f_X}} \,. \quad (59)$$

Therefore, the gate count of the new design A needs to be smaller than $\frac{G_X D_{C_X}}{D_{C_A}} \sqrt{\frac{f_A}{f_X}}$ so that A is better than the best reference design X.

Similarly, if gate count and $PCMK$ are the chosen input/output, and Y is the highest-$PCMK$ reference, we have:

$$G_A < \frac{G_Y K_Y D_{C_Y}^2}{K_A D_{C_A}^2} \frac{f_A}{f_Y} \,. \quad (60)$$

Overall, in order to have the best design in terms of both $PCMG$ and $PCMK$, the gate count of the new design A needs to be smaller than the smaller value between $\frac{G_X D_{C_X}}{D_{C_A}} \sqrt{\frac{f_A}{f_X}}$ and $\frac{G_Y K_Y D_{C_Y}^2}{K_A D_{C_A}^2} \frac{f_A}{f_Y}$.

Similar to the gate count lookup process, speed lookup function will compute the optimal value for speed. In order to have a better design in terms of both $PCMG$ and $PCMK$, the speed of the new design A need to be smaller than the smaller value between $\frac{f_X G_A^2 D_{C_A}^2}{G_X^2 D_{C_X}^2}$ and $\frac{f_Y G_A K_A D_{C_A}^2}{G_Y K_Y D_{C_Y}^2}$ (using $PCMG$ and $PCMK$ metrics, respectively).

5.4 Using PCAS example in FIT/IIT case study

This part presents an example of using PCAS in the FIT/IIT case (Fig. 3). After creating a project, defining all inputs, and generating all output formulas, three references ([9], [10] and [8]) are added with their inputs. Their outputs are then automatically computed and presented as in the table in Fig. 2. The users may want to design an IIT with the shown inputs. If the users estimate their design speed as

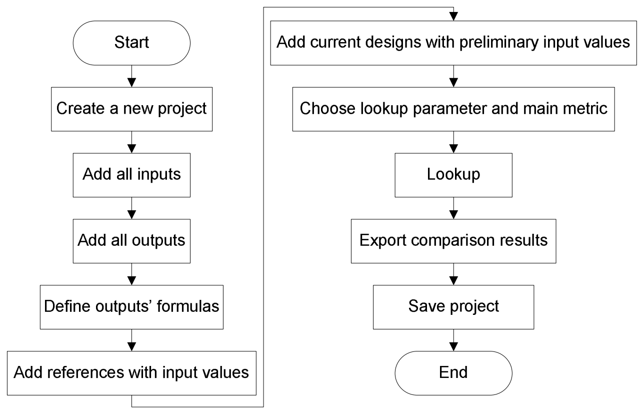

Start
Create a new project
Add all inputs
Add all outputs
Define outputs' formulas
Add references with input values
Add current designs with preliminary input values
Choose lookup parameter and main metric
Lookup
Export comparison results
Save project
End

Figure 3. PCAS example flowchart

200MHz, and choose to lookup for the gate count, PCAS can analyze the references and compute the possible maximum gate counts. The results are 28.54K and 28.84K gates for *PCMG* and *PCMK* metrics, respectively. Thus, in order to have better PCMs than the references, the new design must have the gate count smaller than 28.54K gates.

5.5 Flexible design of PCAS

PCAS is developed with the aim to facilitate the use of PCM technique. Moreover, PCAS is flexibly designed so that it can facilitate different metrics for other architectures. Its flexibility is supported by the followings arguments. Firstly, the inputs and outputs of projects are easily added or removed. Secondly, the formulas of the outputs are modifiable with PCMs as the default formulas. Thirdly, the lookup parameter and the metric can be arbitrarily chosen among the inputs and outputs for the lookup function.

6 Conclusions

In this paper, a Performance-Cost Analysis Software is proposed to facilitate the use of the PCM technique, which is proposed in our other manuscript, for a comprehensive VLSI designs comparisons. Using the software, users can manage the reference designs, generate analyzing formulas, analyze and lookup the allowed boundaries in their designs and export comparison results. PCAS is flexibly designed in order to facilitate the use of not only our PCM technique for FIT/IITs, but also other metrics for other architectures.

References

[1] "Draft ITU-T Recommendation H.264 and Draft ISO/IEC 14496-10 AVC," in Joint Video Team of ISO/IEC JTC1/SC29/WG11 & ITU-T SG16/Q.6 Doc. JVT-G050, Mar. 2003.

[2] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. CSVT*, Vol. 13, No. 7, pp. 560-576, Jul. 2003.

[3] ISO/IEC 14496, "Coding of Moving Pictures and Audio", Mar. 2002.

[4] S. Gordon, D. Marple, and T.Wiegand, "Simplified use of 8×8 transforms – updated proposal and results," Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 11th Meeting, JVT-K028, Munich, Germany, Mar. 2004.

[5] G. Pastuszak, "Transforms and Quantization in the High-Throughput H.264/AVC Encoder Based on Advanced Mode Selection," *Proc. ISVLSI'08*, pp. 203-208, Apr. 2008.

[6] W.J. Choi, J.H. Park, and S.S. Lee, "A High-Performance & Low-Power Unified 4×4 / 8×8 Transform Architecture for the H.264/AVC Codec," *Proc. IVCNZ*, pp. 1-6, Nov. 2008.

[7] N.T. Ngo, T.T.T. Do, T.M. Le, Y.S. Kadam, and A. Bermak, "ASIP-controlled Inverse Integer Transform for H.264/AVC Compression," *Proc. IEEE/IFIP RSP*, pp.158-164, Jun. 2008.

[8] T.T.T. Do and T.M. Le, "High Throughput Area-Efficient SoC-Based Forward/Inverse Integer Transforms for H.264/AVC," accepted for presentation at *ISCAS'10.*

[9] Y-C. Chao, H-H. Tsai, Y-H. Lin, J-F. Yang, and B-D. Liu, "A Novel Design for Computation of all Transforms in H.264/AVC Decoders," *Proc. IEEE ICME'07*, pp. 1914-1917, Jul. 2007.

[10] G.-A. Su and C.-P. Fan, "Low-Cost HW-Sharing Architecture of Fast 1-D Inverse Transforms for H.264/AVC and AVS Applications," *IEEE Trans. CAS II: Expr. Briefs*, Vol. 55, No. 12, pp. 1249-1253, Dec. 2008.

[11] T.T.T. Do and T.M. Le, "A High Normalized Aggregate Throughput SoC-based Inverse Integer Transform Design for H.264/AVC," *Proc. ISIC'09*, pp. 453-456, Dec. 2009.

[12] S. Pasricha and N. Dutt, *On-Chip Communication Architectures - System on chip interconnect*, Morgan Kaufmann, 2008.

[13] J.M. Rabaey, Anantha P. Chandrakasan, and Borivoje Nikolić, *Digital Integrated Circuits: a Design Perspective*, 2nd edition, Prentice Hall, 2003.

[14] D.A. Hodges, H.G. Jackson, R.A. Saleh, *Analysis and design of digital integrated circuits: in deep submicron technology*, McGraw-Hill Science/Engineering/Math, 3rd edition, 2003.

[15] B.S. Landman and R.L. Russo, "On a pin versus block relationship for partitions of Logic Graphs," *IEEE Trans. on Computers*, Vol. c-20, No. 12, pp. 1469-1479, Dec. 1971.

[16] N.H.E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd edition, 2004.

[17] M. Celik, L. Pileggi, and A. Odabasioglu, *IC Interconnect Anal.*, 2002.

[18] D. Sylvester and K. Keutzer, "System-level Performance Modeling with BACPAC - Berkeley Advanced Chip Performance Calculator," *Proc. Workshop System-Level Interconn. Prediction*, pp. 109-114, Apr. 1999.

[19] Standard cell libraries of c35 AMS, 0.13 IBM SF8.

[20] M.D. Ciletti, Advanced Dig. Design with Verilog HDL, Prentice Hall, 2003.