FIFO-level-based Power Management and its Application to an H.264 Encoder

Ngoc-Mai Nguyen, Warody Lombardi, Edith Beigné,
Suzanne Lesecq
Univ. Grenoble Alpes, F-38000 Grenoble, France
CEA, LETI, MINATEC Campus, F-38054 Grenoble, France
{firstName.LastName}@cea.fr

Xuan-Tu Tran
SIS Laboratory, VNU University of Engineering and Technology
144 Xuan Thuy road, Cau Giay, Hanoi, Vietnam
tutx@vnu.edu.vn

***Abstract*— Dynamic Power Management and Dynamic Voltage and Frequency Scaling have been investigated during the last decades to reduce the power consumption of electronic circuits and systems. This paper proposes a new implementation of dynamic frequency scaling to manage the power consumption based on the occupancy level of a FIFO in a communication link between two components. Based on control theory, the method is simple and application independent. The PI controller proposed here was first tested in the MATLAB environment and then designed using VHDL. Simulations in the MATLAB environment and in ModelSim have allowed the validation of the control technique proposed. Being synthesized in technology FD-SOI 28*nm* using RC synthesis tool from Cadence, the control method presents only 10.8% of Silicon area overhead with 1.6% of power consumption reduction. This preliminary result is appealing because it has been obtained without voltage scaling that will improve the power gain even more.**

***Keywords*—power management, FIFO-level-based, PI controller, H.264 encoder**

I. Introduction

Many research efforts have been made to produce electronic circuits and systems that can perform more tasks with limited resources while consuming less energy. The technology development has increased significantly the operating speed and integration density. In the meantime, energy and power consumption have become a tremendous problem because mobile devices are powered by batteries with limited capacity. Moreover, the circuit reliability and cooling cost along with environmental considerations are also related to the power consumption. Static and dynamic techniques have been proposed in order to decrease the power consumption. Static techniques are applied at design time while dynamic ones adapt the circuit functioning during the operating time [1]. Dynamic Power Management (DPM) enables reconfiguration ability to electronic systems in order to provide requested services and performance levels with a minimum number of active components or a minimum workload on such components [2]. DPM can be realized at different levels, from transistor circuits to hardware/software systems. Note that DPM solutions have been investigated for the last few decades [3]. Power consumption in digital CMOS circuits can be modeled as follows [3][4]:

$$\begin{aligned} \mathrm{P_{total}} &= K.C_L.V_{dd}.V_{swing}.f + I_{sc}.V_{dd} + I_{leakage}.V_{dd} \\ &= P_{sw} + P_{sc} + P_l \end{aligned} \quad (1)$$

The main part is the dynamic power P_{dyn} that contains the switching P_{sw} and short-circuit P_{sc} powers. The leakage power P_l depends on the technology at hand. In most cases, the voltage swing V_{swing} is equal to the supply voltage V_{dd} [3]. Hence, P_{sw} can be expressed as the product of the node transition activity factor K, the load capacitance C_L, the square of supply voltage V_{dd} and the clock frequency f. Thus, K, C_L, f and V_{dd} influence the power consumption. Note that acting on V_{dd} is usually preferred because of the quadratic dependence on P_{sw} and its linear dependence on P_{sc} and P_l. Some power reduction approaches tried to reduce the switching activity by re-encoding data, reconstructing the calculation circuits, modify the communication links, memory and hierarchy [4]. Lowering the frequency f also scales down the power consumption. For idle components, it is also possible to stop the clock or enter a power-off mode. However, even in active states, components do not always need to work at their maximum performance. Therefore, f and V_{dd} can vary (e.g. scale down) to operate at lower power states [2].

Dynamic Voltage and Frequency Scaling (DVFS) techniques implement the scaling of both factors. In the literature, power management policies are classified into time-out, predictive and stochastic ones. Time-out policies decide to switch into low-power mode after a certain period of idleness. Predictive and stochastic policies are based on predicted or stochastically modeled workload. Note that workload modeling seems to be the most efficient solution while both time-out and predictive policies present lower efficiency, inaccurate workload prediction and application dependency. The reader can refer to [1] where a summary and comparison of several power management policies is given. Moreover, references [5] to [11] provide new solutions and improvements of Dynamic Power Management techniques.

The main objective of the present paper is to manage the power consumption by scaling the frequency according to the status of a FIFO link between components. Note that adding voltage scaling has a strong impact on the power consumption reduction, as can be seen in (1). The control method is simple and application-independent so that it can be widely implemented in FIFO-based systems. For instance, multimedia applications (e.g. video, audio codec) often require a long computational path, usually implemented in stages with buffers in-between. Data is written to and read from these buffers in the same order, which is similar to a FIFO. Note that the control must be simple enough to be implemented in hardware (i.e. directly in Silicon). This strong constraint makes most of the advanced control techniques intractable.

The FIFO control developed here will be embedded on a hardware H.264 encoder. The "Video Encoder for the Next Generation Multimedia Equipment" (VENGME) design is an H.264/AVC encoder targeting mobile platforms [12]-[14]. Its architecture, as illustrated in Fig. 1, is a pipeline one with buffers (blocks with dots) placed between modules. Data in the current frame is fed into the chip via an AHB/APB interface. Then, it is written in current macroblock RAM (Cur MB RAM) and search window RAM (SW RAM). The predicted data, from intra prediction (Intra Pred) and inter prediction, containing integer motion estimation (IME), fractional motion estimation (FME) and motion compensation (MC), are stored in buffers placed inside these modules before being sent to the forward transformation and quantization (FTQ) block. TQIF is a memory block implemented as an interface to transfer data from FTQ to the de-quantization and inverse transformation (ITQ) block and the entropy coder (EC). The entropy encoded data, after being encapsulated in the network abstraction layer (NAL), are stored in NAL SRAM before being sent out via off-chip interface. The same applies to data from de-blocking filter (DF). In the decoding path, the reconstructed frame (from REC module) is stored in the Intra Ref RAM associated to the Intra Pred block.

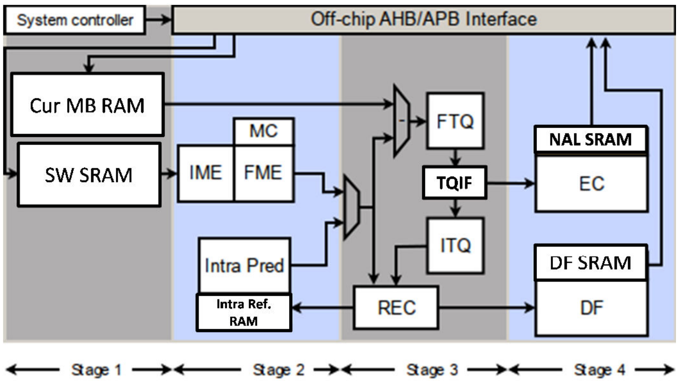


Fig. 1. VENGME H.264/AVC encoder architecture.

The control technique for power reduction is applied to the EC module that contains the entropy coder communicating with the video data packer at the network abstraction layer (VDP NAL) via a FIFO [13]. From a deep analysis of the workload and computing speed of the different blocks, it has been seen that having a fine grain adjustment of the clock frequency (and supply voltage) on the NAL block will decrease the power consumption. Actually, the clock frequency can be scaled down when the FIFO is far from full, leading to power consumption decrease (1).

Moreover, design constraints make the actual H.264 platform "splitable" in two main parts, namely, the NAL block on one side, and the other blocks on the other side. As a consequence, the platform is split in two (voltage-) frequency domains, each one having its own adaptable clock frequency. Therefore, it is natural to implement FIFO control based on control theory between the EC and NAL blocks. Lastly, the output of VDP NAL is a memory block with no timing constraint.

The rest of the paper is organized as follows. Section II reviews previous power optimization works from the literature. The works reported here are based on buffer information. Section III summarizes the problem to be solved and its modeling aspects. Then, the control design is presented. Hardware implementation and simulation results are reported in Section IV. Finally, Section V concludes the paper and gives future works directions.

II. FIFO-Based Power Management Methods

A FIFO-based system can be described as a system with modules that communicate via FIFO links. For each FIFO link, a "producer" module writes data into the FIFO when it is not full. The "consumer" module reads data in the same order the data is written until the FIFO is empty. Fig. 2 illustrates this communication method. To enable Dynamic Voltage and Frequency Scaling, producer and consumer operate with independent supply voltages V_{dd} and clock frequencies f (f_P and f_C). Hence, the integrated FIFO must be an asynchronous one [15].

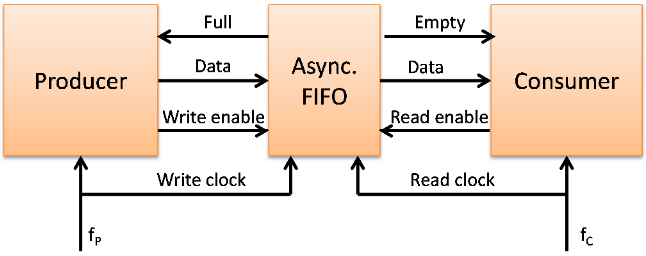


Fig. 2. FIFO communication between producer and consumer.

In the literature, some researches tried to control V_{dd} and/or f based on the FIFO/buffer status. For instance, L. Thiele *et al.* [16] used the buffer fill-in level that indicates the data stream rate combined with static worst-case bounds based on workload history to adapt the clock rate of the processor (producer). In [15], P. Choudhary *et al.* compared the time that producer and consumer have to wait due to FIFO fullness or emptiness in a given time interval T_{sample} to decide if it is necessary to match the rate between the frequencies of both domains. The calculated rate is used to scale either the frequency of the producer or of the consumer according to the system constraint. However, one has to wait until the FIFO is full or empty for a time interval long enough before acting on f_C and/or f_P.

Several works proposed the use of buffer occupancy to perform DVS or DVFS [17]-[20]. Y-H. Lu et al. [17] inserted buffers in a multimedia system. A graph-walk algorithm was implemented to assign the processor (producer) frequency based on the buffer state. In the assignment graph, each vertex contains information on the processor frequency, the amount of data in the buffer and the next operation (writing or reading data to or from the buffer). The methods developed in [18] and [19] were applied to a processor operating as a multimedia decoder and communicating to a display device via a buffer. A closed-loop control system was implemented to adjust the decoder speed in order to keep the buffer occupancy within a given safe range so as to match decode and display rates. However, [17]-[19] FIFO/buffer control approaches were designed for specific applications and implemented in software. Similarly, in [20], a finite state machine (FSM) was implemented to increase/decrease the frequency and voltage by one single step. The voltage/frequency changes are decided based on the queue signal compared to a deviation window. The queue signal is either the difference between the queue occupancy and a reference value or the difference between the queue occupancy at two consecutive sampling times while the deviation window is the threshold determined for the queue signal.

III. System Modeling and Control Design

A. System modeling

As mentioned in Section I, the method proposed has been implemented in the EC module of an H.264 encoder where the encoder acts as "producer", sending data to the VDP NAL which is the "consumer", see Fig. 3. The EC is the last module in the encoding path. The so-called producer receives data from the other modules in the VENGME H.264 encoder. Hereafter, f_P (applied to all the blocks in the dashed rectangle) is supposed constant while f_C can vary.

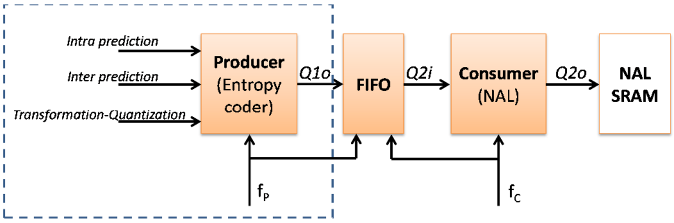


Fig. 3. Entropy coder module to apply power management control method.

The FIFO can be modeled as a tank with one input Q1 and one output Q2 extracted using a "pump" (see Fig. 4(a)). However, the FIFO data is discrete, given in packets (Fig. 4(b)). As a consequence, the FIFO dynamic model is given by:

$$FIFO(k+1) = FIFO(k) + Q1(k) - Q2(k) \qquad (2)$$

where *FIFO(k)* is the occupancy level of the FIFO in number of packets at the $k^{\text{-th}}$ sampling time. *Q1(k)* is the number of data packets entering the FIFO from the producer. *Q2(k)* is the number of data packets exiting the FIFO to feed the consumer. Due to Silicon area constraints for the H.264 hardware platform, the maximal number of packets in the FIFO is chosen equal to 7. Therefore, $FIFO(k) \in \{0, 1, 2, 3, 4, 5, 6, 7\}$.

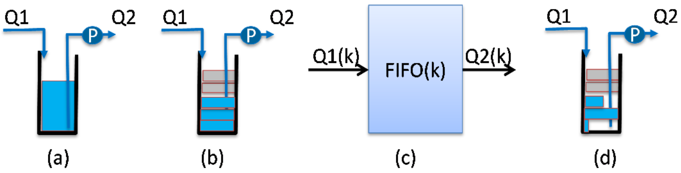


Fig. 4. FIFO model construction.

The actual number of packets in the FIFO also depends on the clock frequency of the consumer f_C. Moreover, it also depends on the length of the packets as packets are of different length depending on the data at hand; see Fig. 4(d). However, for the sake of simplicity, the FIFO output flow $Q2$ is supposed proportional to $f_C(k)$:

$$Q2(k) = bf_C(k) \qquad (3)$$

where b is a positive constant.

To identify the constant b, the VENGME encoder encodes benchmark video frames to trace the data flow and the FIFO status. The FIFO status contains the producer/consumer stall state similar to the one in [15], but not only full/empty as normal. From this data, when the consumer does not have to wait for data available in the FIFO, i.e. when the consumer is not stall, b is equal to $20ns$.

Each time the FIFO is full (resp. empty), and the producer has some data to write into the FIFO (resp. the consumer wants to read data from the FIFO), the producer (resp. consumer) falls into the stall state. This latter state leads to a waste of power consumption. Thus, the control objective is to adapt the consumer frequency to keep the FIFO half-full (neither empty nor full) during normal operation. Of course, at the end of the encoding process, the FIFO will have to be emptied.

B. Control design

For the FIFO link under study, the input data $Q1$ is the output of the producer, which is not controlled. Therefore, in the system model, $Q1$ is considered as a disturbance. The FIFO transfer model, without disturbance, in the z-domain, is as follows:

$$\frac{FIFO(z)}{Q2(z)} = G(z) = \frac{-1}{z-1} \qquad (4)$$

which is actually an integrator. Fig. 5 shows the controlled system with its controller *C(z)* that adapts the consumer frequency $f_C(z)$ to balance the FIFO level at a Reference value. The input of the controller is the error *E(z)* between the Reference level and the FIFO level.

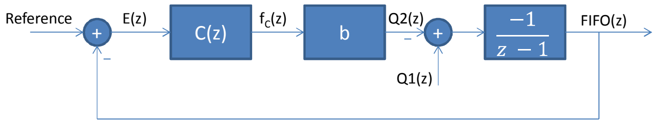


Fig. 5. Closed-loop control scheme of the FIFO level.

A discrete-time Proportional-Integral (PI) controller [21] is selected to reject the "disturbance" $Q1$, to ensure a closed-loop functioning without static error and to tune the closed-loop system time response. The controller is modeled as:

$$\frac{f_C(z)}{E(z)} = C(z) = K_p + K_i\frac{z}{z-1} \qquad (5)$$

Therefore, the closed-loop transfer function is given by:

$$\frac{FIFO(z)}{R(z)} = \frac{C(z)b\frac{-1}{z-1}}{1+C(z)b\frac{-1}{z-1}} \qquad (6)$$

$$\frac{FIFO(z)}{R(z)} = \frac{-[(K_p+K_i)z - K_p]b}{z^2 - z[2+b(K_p+K_i)] + 1 + K_p b} \qquad (7)$$

where R is the Reference for the FIFO level.

The poles z_1, z_2, determine the system dynamic characteristics in closed-loop. Their numerical values depend on K_p and K_i values. Once the control designer has chosen z_1 and z_2, K_p and K_i are computed with:

$$K_p = \frac{z_1 z_2 - 1}{b} = \frac{z_1 z_2 - 1}{0.02} \tag{8}$$

$$K_i = \frac{z_1 + z_2 - z_1 z_2 - 1}{b} = \frac{z_1 + z_2 - z_1 z_2 - 1}{0.02} \tag{9}$$

where b = $0.02\mu s$ according to its identification given above.

C. *Simulation results using MATLAB*

From the data in the simulation trace, the system and controller have been modeled using the MATLAB environment. In this simulation, the maximal FIFO level is 7 and the reference level is 2. The consumer frequency is taken in the set of values {0, 12.5, 25, 50, 100} (MHz). This discrete set of values for f_C adds non-linearities in the system. Note that these non-linearities are not considered in the present paper.

The choice of the closed-loop poles is a strategic because they will impose the closed-loop dynamics. Firstly, a fast closed-loop dynamics response is considered and the poles $z_{1,2}$ = 0.5 ± 0.2i are chosen. Fig. 6 shows the closed-loop behavior of the FIFO level (left) together with the clock frequency value (right). Even if the behavior is satisfactory (taking into account the disturbance Q1), from an implementation point-of-view, this pole choice will imply strong constraints on the clock frequency engine (i.e. actuator) that delivers f_C. Actually, this latter will react as soon as a change in the system is detected, leading to many changes in the clock frequency engine output. Note that these fast changes will add extra power consumption.

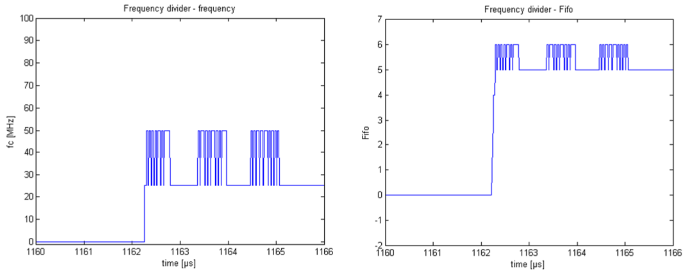


Fig. 6. Simulation results for closed-loop poles $z_{1,2}$ = 0.5 ± 0.2i (right: FIFO level, left: clock frequency value f_C).

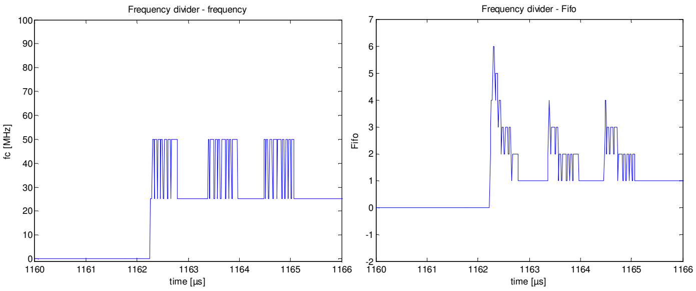


Fig. 7. Simulation results for closed-loop poles $z_{1,2}$ = 0.9 (right: FIFO level, left: clock frequency value f_C).

As a consequence, another pole selection is proposed ($z_{1,2}$ = 0.9) in order to slow down the closed-loop dynamics and impose less constraints on the clock frequency engine. The simulation results are shown in Fig. 7. As expected, the behavior is consistent with the control requirements while the clock frequency engine supports less dynamic constraints.

Both simulation cases show the effectiveness of the PI control of the FIFO level. The controller must now be implemented in hardware (i.e. in Silicon, not in Software). This Silicon implementation will impose extra constraints in order to limit the Silicon area of the controller.

IV. Hardware Implementation and Results

A. *Architecture*

The controller designed in the previous section is implemented in hardware, within the chip. From the controller transfer function (in the z-domain)

$$\frac{f_C(z)}{E(z)} = K_p + K_i \frac{z}{z-1} \tag{10}$$

$$z f_C(z) - f_C(z) = \left(K_p + K_i\right) z E(z) - K_p E(z) \tag{11}$$

the recurrence equation is derived:

$$f_C(k+1) - f_C(k) = \left(K_p + K_i\right) E(k+1) - K_p E(k) \tag{12}$$

which is equivalent to:

$$f_C(k) = f_C(k-1) + \left(K_p + K_i\right) E(k) - K_p E(k-1) \tag{13}$$

A calculating circuit can be constructed from (13) to calculate the consumer frequency $f_C(k)$ from $f_C(k\text{-}1)$ and from the errors $E(k)$ and $E(k\text{-}1)$. This calculating circuit contains two multipliers with constants ($K_p + K_i$) and K_p.

To avoid implementing multipliers with too large coefficients, the calculating circuit is modified as follows:

$$f_{sel}(k) = f_{sel}(k-1) + \frac{K_p + K_i}{12.5} E(k) - \frac{K_p}{12.5} E(k-1) \tag{14}$$

where f_{sel} is defined as the value to select the corresponding consumer frequency, which is equal to $\left(\frac{1}{12.5}\right) f_C$. Fig. 8 illustrates the calculating circuit.

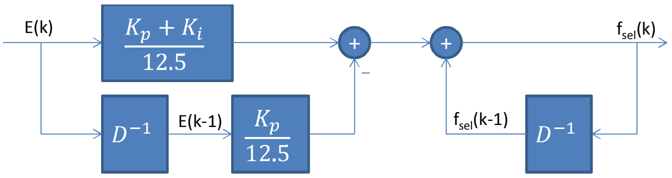


Fig. 8. Calculating circuit to be implemented.

The hardware implementation is easier if the closed-loop poles are chosen so that the coefficients of the multipliers are multiples of (1/8). In this way, the latter will require at most only three bits in the fractional part of the binary expression.

The closed-loop poles must also ensure the expected performances for the closed-loop system (included the stability). As shown in the simulation performed in the MATLAB environment, having poles with small absolute values will introduce strong dynamic constraints on the clock frequency engine. Therefore, the poles that will be selected have to limit the overload (in terms of fast changes) on the frequency engine. Moreover, real poles will be preferred in order to avoid pseudo-periodic behaviors for the output system. Taking into account the range [0.5, 1) for the closed-loop poles, 21 different pairs of poles that satisfy all the constraints above can be selected.

All the 21 corresponding controllers have been implemented in hardware. Fig. 9 presents the architecture of such a controller, whose poles are $z_1 = 0.75$ and $z_2 = 0.5$. Hence $K_p = -31.25$ and $K_i = -6.25$ and f_{sel} is given by:

$$f_{sel}(k) = f_{sel}(k-1) - \frac{6}{2}E(k) + \frac{5}{2}E(k-1) \tag{15}$$

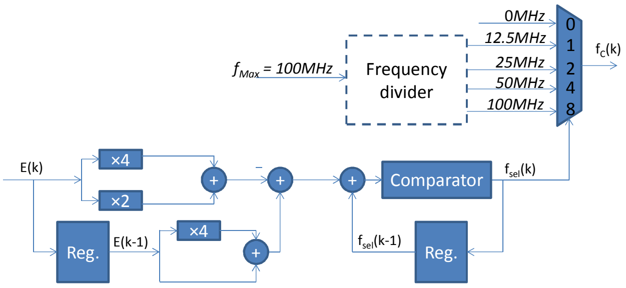


Fig. 9. Architecture of the controller when $z_1 = 0.75$ and $z_2 = 0.5$.

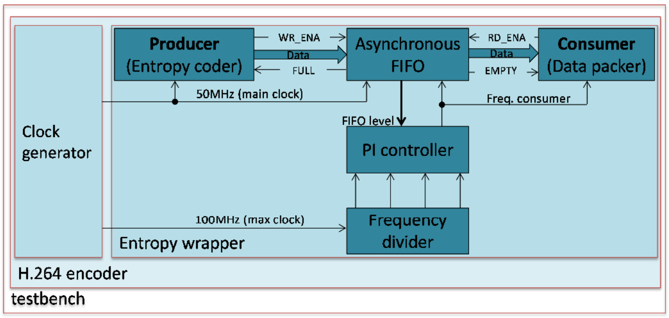


Fig. 10. Integration of the DFS controller in the H.264 encoder.

The implemented calculating circuit contains only four adders and two registers. The multiplications by 2 and 4 are simply done by bit-shifting. To perform frequency non-uniform quantizer, a comparator and multiplexor are added to select the corresponding frequency from the available set of clock signals. These signals are given from a frequency divider implemented using a four-bit counter. Note that to save Silicon area, it is preferable for the calculating circuit to have small fractional part in binary expression and few adders.

Each controller is then integrated into the VENGME video encoder in order to evaluate its performance within the encoder, when a benchmark of videos is run. The integration of the controller is described in Fig. 10.

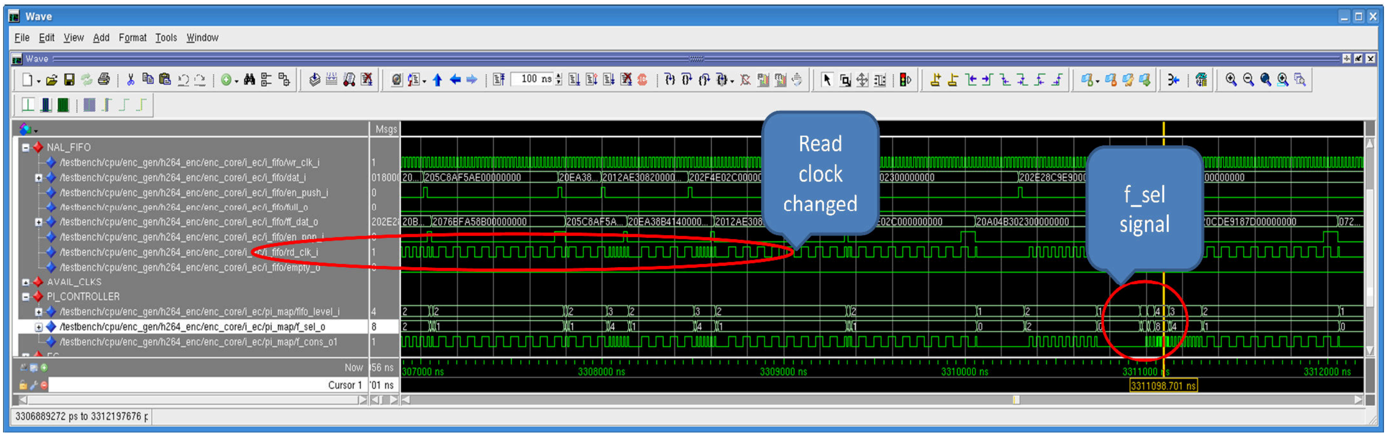
In the VENGME H.264 encoder, a clock generator whose maximal frequency generated is 100*MHz* is already implemented. This clock signal is then used as the input of the frequency divider. The main clock (50*MHz*) is used for most of the modules in the H.264 encoder. The next sub-section presents simulation and power estimation results.

B. *Results*

1) Simulation results

When the FIFO occupancy level reference value is changed, a trade-off between the power consumption and the computing performance is performed [20]. In the hardware simulation, the reference value is chosen equal to 3. The simulation results prove again the impact of the closed-loop pole value. For controllers with the module of the associated closed-loop poles close to 1, the consumer frequency and FIFO level change gradually. For instance, for closed-loop poles equal to $z_1 = 0.875$ and $z_2 = 0.75$, the maximal frequency of 100*MHz* is not reached, even when the FIFO is full for a long time. The consumer frequency jumps up one step at once. With the controller whose associated poles are $z_{1,2} = 0.5$, the FIFO is rarely full and the consumer frequency can jump from 0*MHz* to 50*MHz* in just one sampling time.

Fig. 11 shows simulation results when the controller reacts in a smooth way and the FIFO is rarely full. This selection is also good in terms of hardware implementation. It costs only four adders and one bit for the fractional part. The closed-loop pole values are $z_1 = 0.75$ and $z_2 = 0.5$. Hence, $K_p = -31.25$ and $K_i = -6.25$. The simulation is performed using ModelSim from Mentor Graphics. The simulation waveforms show that while the clock frequency on the "write side" of the FIFO (i.e. producer frequency) keeps unchanged, the clock frequency on the "read side" (consumer frequency) varies according to the control decision. The control decision can be seen in the figure as the values of signal f_{sel} (0, 1, 2, 4, 8) that correspond with the frequency values equal to 0, 12.5, 25, 50 and 100*MHz*.



Wave
File Edit View Add Format Tools Window
Wave
Msgs
NAL_FIFO
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/wr_clk_i
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/dat_i
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/en_push_i
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/full_o
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/ff_dat_o
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/en_pop_i
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/rd_clk_i
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/i_fifo/empty_o
AVAIL_CLKS
PI_CONTROLLER
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/pi_map/fifo_level_i
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/pi_map/f_sel_o
/testbench/cpu/enc_gen/h264_enc/enc_core/i_ec/pi_map/f_cons_o1
Read clock changed
f_sel signal
Now
Cursor 1
307000 ns
3308000 ns
3309000 ns
3310000 ns
3311000 ns
3312000 ns
3311098.701 ns
3306889272 ps to 3312197676 p

Fig. 11. Waveform of the selected PI controller applied onto the system.

2) *Power and energy estimation results*

The proposed control method needs the implementation of an asynchronous FIFO and the PI controller that will add extra Silicon. For FD-SOI 28nm technology, the RC synthesis tool from Cadence shows a Silicon area overhead of 10.8% when compared to the original EC module. Note that in our application, DFS is applied only on the VDP NAL part, i.e. the consumer. Because it is a very small block, the power consumption gain on the total EC module is only 1.6%. This latest result is obtained without V_{dd} scaling. Thus with DVFS, the power consumption can be decreased even more because $P_{dyn} \sim f V_{dd}^2$. In the case where the closed-loop poles are $z_{1,2} = 0.5 \pm 0.2i$, the dynamic part of the energy possible gain on the consumer side, is evaluated in the MATLAB environment thanks to an integration of the consumer clock frequency f_C over a time period of $0.1s$. When compared to the solution without FIFO level control, the gain is equal to 96.8%. Again, voltage scaling will improve this gain.

V. Conclusions

In this paper, a method to scale the clock frequency based on control theory is proposed. The control input is the FIFO occupancy level. A controller scheme that is simple, possibly implemented in hardware and independent from the application is proposed. A FIFO-based system containing FIFO links, each link connecting producer and consumer, was firstly modeled. A PI controller was designed and implemented to adapt the consumer frequency according to the FIFO occupancy level. The coefficients of the controller were selected to meet the required performances for the closed-loop system, taking also into account hardware implementation constraints. Synthesis results using the RC tool from Cadence show a Silicon area overhead of 10.8% with a decrease in the power consumption of 1.6%. This latest result is very appealing because it has been obtained without voltage scaling: with a complete DVFS scheme, the power gain will be even larger.

Future works include the extension of this FIFO control approach to the whole VENGME architecture. Voltage scaling technique will be as well integrated in the platform. Post-synthesis power estimation results obtained using PrimeTime are currently under analysis to evaluate in a finer way the efficiency of the FIFO control method proposed here.

Acknowledgment

This work is partly funded by the Vietnam National University, Hanoi (VNU) through research project number. QGDA.10.02 (VENGME), projects Catrene HARP number CA112 and Catrene BENEFIC number CA505. This work was also in the scientific frame of the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

References

[1]. Yung-Hsiang Lu; De Micheli, G., "Comparing system level power management policies," IEEE Design & Test of Computers, vol.18, no.2, pp.10,19, March/April 2001.

[2]. Benini L.; Bogliolo A.; De Micheli G., "A survey of design techniques for system-level dynamic power management", IEEE Tr. on Very Large Scale Integration (VLSI) Systems, vol.8, no.3, pp.299-316, June 2000.

[3]. Chandrakasan, A.P.; Sheng, S.; Brodersen, R.W., "Low-power CMOS digital design", IEEE Journal of Solid-State Circuits, vol.27, no.4, pp.473,484, April 1992.

[4]. Li-Chuan Weng; XiaoJun Wang; Bin Liu, "A survey of dynamic power optimization techniques," 2003 IEEE 3rd International Workshop on System-on-Chip for Real-Time Applications, pp.48,52, July 2003.

[5]. Stangaciu, C.S.; Micea, M.V.; Cretu, V.I., "Energy efficiency in real-time systems: A brief overview", 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp.275-280, 23-25 May 2013.

[6]. Zhuravlev, S.; Saez, J.C.; Blagodurov, S.; Fedorova, A.; Prieto, M., "Survey of Energy-Cognizant Scheduling Techniques", IEEE Trans. on Parallel and Distributed Systems, vol.24, no.7, pp.1447-1464, July 2013.

[7]. Kihwan Choi; Soma, R.; Pedram, M., "Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.24, no.1, pp.18,28, January 2005.

[8]. Gang Qu, "Power Management of Multicore Multiple Voltage Embedded Systems by Task Scheduling," Parallel Processing Workshops, 2007. ICPPW 2007, pp.34-34, 10-14 September 2007.

[9]. Sulaiman, D.R., "Using clock gating technique for energy reduction in portable computers", International Conference on Computer and Communication Engineering - ICCCE 2008, pp.839-842, May 2008.

[10]. Yadav, M.K.; Casu, M.R.; Zamboni, M., "DVFS Based on Voltage Dithering and Clock Scheduling for GALS Systems", 2012 IEEE 18th International Symposium on Asynchronous Circuits and Systems (ASYNC), pp.118-125, 7-9 May 2012.

[11]. Herbert, S.; Garg, Siddharth; Marculescu, D., "Exploiting Process Variability in Voltage/Frequency Control", IEEE Trans. on VLSI Systems, vol.20, no.8, pp.1392-1404, August 2012.

[12]. Xuan-Tu Tran; Van-Huan Tran, "Cost-efficient 130nm TSMC Forward Transform and Quantization for H.264/AVC encoders", 2011 IEEE 14th Int. Symp. on Design and Diagnostics of Electronic Circuits & Systems (DDECS), pp.47-52, 13-15 April 2011.

[13]. Ngoc-Mai Nguyen; Beigne, E.; Lesecq, S.; Vivet, P.; Duy-Hieu Bui; Xuan-Tu Tran, "Hardware implementation for entropy coding and byte stream packing engine in H.264/AVC", 2013 International Conference on Advanced Technologies for Communications (ATC), pp.360-365, 16-18 October 2013.

[14]. Nam-Khanh Dang, Xuan-Tu Tran, Alain Merigot, "An Efficient Hardware Architecture for Inter-Prediction in H.264/AVC Encoders", 2014 IEEE 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), 23-25 April 2014.

[15]. Choudhary, P.; Marculescu, D., "Power Management of Voltage/Frequency Island-Based Systems Using Hardware-Based Methods", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, , vol.17, no.3, pp.427-438, March 2009.

[16]. Thiele, L.; Chakraborty, S.; Maxiaguine, A., "DVS for buffer-constrained architectures with predictable QoS-energy tradeoffs", Third IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS'05, pp.111-116, September 2005.

[17]. Yung-Hsiang Lu; Benini, L.; De Micheli, G., "Dynamic frequency scaling with buffer insertion for mixed workloads", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, , vol.21, no.11, pp.1284-1305, November 2002.

[18]. Zhijian Lu; Lach, J.; Stan, M.; Skadron, K., "Reducing multimedia decode power using feedback control", 21st International Conference on Computer Design, pp.489-496, 13-15 October 2003.

[19]. Geuntae Bae; Jaesub Kim; Daewon Kim; Daeyeon Park, "Low-power multimedia scheduling using output pre-buffering", 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp.389-396, Sept. 2005.

[20]. Wu, Q.; Juang, P.; Martonosi, M.; Clark, D.W., "Voltage and frequency control with adaptive reaction time in multiple-clock-domain processors", 11th International Symposium on High-Performance Computer Architecture – HPCA'11, February 2005.

[21]. Astrom, K. J., and Björn Wittenmark. "Computer controlled systems: theory and design", 3rd edition, Prentice Hall, 1986.