

Behavioral-level event-driven power management for DECT digital receivers

N.D. Zervas^{a,*}, G. Theodoridis^b, D. Soudris^c

^a*ALMA Technologies, Marathonos Av. 2, Pikermi-Attika, 19009 Greece*

^b*Section of Electronics and Computers, Department of Physics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*

^c*VLSI Design and Testing Center, Dept. of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece*

Received 1 August 2003; received in revised form 7 August 2004; accepted 4 October 2004

Abstract

Power management is a low-power design technique applicable in almost all design levels. Here, the idea of exploiting events to trigger the shut-down of hardware resources is applied at the behavioral-level of a DECT digital receiver design. Power management involves a trade-off between the power savings arising from the power-down (or shut-down) parts of the system and the power increase due to the additional logic for the generation of the shutdown signals. For that purpose, taking into account the digital receiver's characteristics, a behavioral-level power management technique is introduced. The efficiency of the proposed technique is proven by its application on an industrial DECT receiver, where a power saving of 50% in terms of the dynamic power consumption is achieved.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Behavioral power management; Low-power design; Power estimation; Shut-down techniques; Digital receivers

1. Introduction

In the last few years, there is a continuously growing demand for wireless terminals integrating sophisticated multi-service applications. Wireless multi-service terminals based on the DECT standard [1], which is a sophisticated platform able to support applications such as voice, fax, and data communications for geographically confined indoor and outdoor areas, have been used widely in the recent years. One of the most challenging problems regarding with the performance and power consumption of a DECT-based wireless system is the design of the baseband part of the receiver, which is directly related with the baseband functionality (i.e. detection, synchronization, and frequency offset correction) [2]. Moreover, the state-of-the-art technology on portable communications imposes strict constraints on the power consumption and area. Thus, the development of low-power, area-efficient strategies are of

critical importance, especially in the high-levels of the design flow where the most significant savings can be achieved [3].

Dynamic power management is one of the most efficient low-power techniques applicable at all the design levels [4]. The basic concept is to shut down parts of the circuit during the time intervals that they perform useless operations. This is achieved by inserting extra logic, the role of which is twofold: (i) to detect the parts of the circuit that perform useless operations and (ii) to generate the appropriate signals to shut down the corresponding hardware components. The shut down procedure is performed either by dropping down or cutting off the supply voltage or by disabling the clock from the corresponding hardware resources. Since additional circuit is inserted, particular attention is needed to preserve the performance of the modified circuit in terms of area, time, and power.

In recent years, power management techniques have been presented in the RT- and logic-level [5–7] as well as in the system-level [8]. In particular, a gated clock technique for sequential circuits has been presented in [5]. The key idea is that during the operation of a Finite State Machine

* Corresponding author.

E-mail addresses: zervas@alma-tech.com (N.D. Zervas), dsoudris@ee.duth.gr (D. Soudris).

(FSM) there exist conditions where the state lines and the output of the FSM are not changed. Clocking the circuit in the corresponding time intervals, results in wasted power consumption both in the combinational logic and registers. Thus, by detecting the idle conditions and by stopping the clock in the corresponding time intervals significant power savings can be obtained.

The power optimization technique of [6] is based on the selective precomputation of the output logic values of a circuit one clock cycle before they are needed to be computed. Using the pre-computed values the internal switching activity of the combinational logic is reduced during the next clock cycle. The switching activity reduction is obtained by freezing part of the primary inputs of the combinational logic in the successive clock cycle.

The approach of [7] is based on placing transparent latches with an enable signal at the input of each block of the circuit, which should be power managed. The idea is to determine, on per clock cycle, those parts of the circuit that perform computations, whose results are used (i.e. will be observable in the output) and those parts of the circuit that perform computations, whose results are useless regarding the whole circuit operation. When a logic block should execute some useful computations, the enable signal makes the latches transparent. Otherwise, the latches retain their previous values blocking any transition within the logic block. In [8], system level power management techniques are reviewed. The basic idea is to predict at run-time the duration of the future idle states of the system based on the system's history. Based on this prediction, it is decided whether it is power- and/or time-efficient to shut down a resource.

However, it seems that none of these techniques has addressed the power management in the behavioral level, where the design of digital receivers usually starts [9]. Moreover, if the power management is postponed for the next design-levels, then it is possible for some power-management options to be disabled.

2. Contribution of the paper

In this paper, a technique consisting of a number of steps applied in a systematic manner for the behavioral-level event-driven power management of DECT digital receivers is introduced.

We have identified special properties appeared in DECT digital receivers, whose exploitation results in an efficient power management technique. Specifically, there are clusters of operations that are executed in a deterministic way in each frame transmission. Furthermore, some of these clusters such as the Automatic Frequency Correction (AFC) should be activated for certain time periods during the frame's transmission. These clusters can be easily identifying either by the specification of the algorithm or by a behavioral description of the algorithm by high-level tools

such as Matlab. Also, there are event-driven signals or we can easily generate such signals to activate and de-activate the identified clusters. By performing power management on these clusters significant power savings can be achieved.

Considering the above, we introduce a power-management technique consisting of three steps: (i) the behavioral analysis, (ii) the extraction of power management scenarios and (iii) the cost function evaluation. The steps are applied for designing power-aware DECT digital receivers in a systematic manner.

The proposed technique is valid for applications/ algorithms, which can be described by acyclic graphs [10]. Indeed, I and Q stream processing do not imply any kind of feedback loop between the inputs I and Q and the output. From hardware point of view, this acyclic structure allows pipelining processing that offers superior performance. The goal of the proposed technique is to determine an optimal power management scenario under which the power savings after the shut down of a resource should be larger than the power consumption increase due to the presence of the additional logic. Also, an exploration of the area is performed. For this reason a behavioral area-power model is proposed and an appropriate cost function is introduced.

The proposed technique consists of the following steps:

- (i) the behavioral analysis of the application to identify the candidate computational clusters at the behavioral level on which the power management can be performed,
- (ii) the extraction of all power management scenarios based on the identified clusters; for that purpose the notion of the event graph is introduced, and
- (iii) the evaluation of the cost function to determine the optimal scenario.

The presented technique is applied to the design of a digital DECT receiver where a power saving of 50% with less than 5% area overhead and no delay penalty, are achieved.

Although, we apply the introduced technique to design a power-aware DECT digital receiver, its applicability is wider. For instance, digital receivers based on the direct conversion architecture, especially those that demodulate Gaussian Minimum Shift Keying (GMSK) signals, could be optimized using the proposed technique. With the direct conversion architecture the receiver's front-end complexity is minimized, since the Intermediate Frequency (IF) down-converting stages are eliminated. This favors digital detection schemes based on In-phase (I) and Quadrature (Q) channel demodulation, which result to receivers structures analogous to the one studied in this paper.

The proposed technique is also applicable in receivers embedding the user-authorization and/or the user-authentication functionalities. In such cases, part of the receiver should operate until a transmission from an authorized user is detected; once an authorized user is detected the user

authentication unit should be ‘turned-off’. Receiver could return to the authorized-user detection phase, if the transmitting side succeeds to correctly identify as specified by the protocol. Finally, to handle cyclic graphs, the method of unrolling the graph could be exploited allowing the application of the proposed technique.

The rest of this paper is organized as follows: in Section 3 the used power model is described. Section 4 describes in detailed manner the steps of the proposed technique, while in Section 5 we describe the DECT digital receiver. Section 6 presents the application of the proposed technique a digital DECT receiver design. Finally, Section 7 summarizes the paper.

3. Power model

The dynamic power dissipation forms the dominant part of the total power in current CMOS technologies and can be expressed by the following formula:

$$P_{\text{dyn}} = \sum_{i=1}^N C_{\text{load}_i} V_{\text{dd}}^2 f E_i \quad (1)$$

where C_{load_i} is the load capacitance at node i , V_{dd} is the power supply voltage, f is the clock frequency and E_i is the activity factor of i -th node. The product $f E_i$ of Eq. (1) is actually the number of transitions from the high-to-low logic value per clock cycle of i -th node. Also, it equals to the ratio of the node transitions from high-to-low logic value over the total number of input vectors:

$$f E_i = f_{1 \rightarrow 0} = \frac{\#\text{trans}(1 \rightarrow 0)_i}{\#\text{vectors}} \quad (2)$$

Substituting (2) into (1), we obtain that:

$$P_{\text{dy}} = \frac{V_{\text{dd}}^2}{\#\text{vectors}} \sum_{i=1}^N C_{\text{load}_i} \#\text{trans}(1 \rightarrow 0)_i \quad (3)$$

Considering the above formula, the power estimation problem is actually a two-dimensional problem, since both the number of transitions and load capacitance have to be estimated. However, in the behavioral level the circuit structure is not fixed yet. Therefore, approximations are taken place during the power estimation. In this paper the switching activity (i.e. $f E_i$) is estimated by calculating the Hamming-distance at the input/output nodes of basic operations (e.g. addition, multiplication, etc.), during the functional simulation. For the capacitance estimation the models presented in [11] are used.

The above power model is adopted due to its low complexity. Considering that the proposed technique is applied at the high levels of design exploration, where additional design parameters (e.g. area, performance) are considered, the used power model should be characterized by low complexity. Thus, we estimate only the dynamic power dissipation, but not the remaining components of

power consumption. Of course, such a power model does not lead to accurate power dissipation values. However, what is required in high-level power exploration is a power model that offers ‘accurate’ values from quantitative perspective rather an accurate power model.

Thus, for that purpose Hamming distance of the input/output signals and the power models of [11] are used to estimate the dynamic power dissipation. The models of [11] consist of simple mathematical formulas and use the Hamming distance of the input/output signals along with pre-characterized capacitance values as free variables to evaluate the power dissipation of a hardware resource. As the mathematical formulas of models of [11] are simple, the complexity of evaluating these formulas is reduced, while the gathering of the values of the free variables (Hamming distances) is accomplished easily and with low complexity by performing functional simulation of the whole design.

Furthermore, we justify this accuracy of our power model in the experimental results comparing the power dissipation derived by the used power model and the power consumption derived after performing logic synthesis on the whole design of DECT receiver. It can be seen from Fig. 6 that the corresponding error of the estimated power dissipation is below of 20%, which is by far adequate for the purposes (power management at the behavioral level) of the adopted power model.

4. Proposed technique

In this section the proposed behavioral-level power management technique for digital receivers is described. For clarity reasons, some definitions are given first:

- Consider a behavioral level description partitioned to a number of behavioral clusters. We denote the *behavioral clusters’ set* as $C = \{c_i | i=0,1,\dots,m-1, m \in \mathbb{N}\}$, where \mathbb{N} is the set of the physical numbers and $m = \|C\|$ is the total number of the behavioral clusters, where $\|\bullet\|$ denotes the cardinality of a set.
- *Event* is defined as an executing behavioral cluster.
- *System period*, T_{SYSTEM} , is the minimum fraction of time during which a sequence of events is not repeated.
- *Event window*, $EW_{i,j}$, is the fraction of system period that lies between the events e_i and e_j .

The proposed event-driven power-management technique is based on the fact that the unobservability of a circuit node at the behavioral level is introduced after the occurrence of an event. A system behavior is a result from a collection of interrelated functions. For instance, MPEG2 application requires, among others, the execution of vector quantization and Huffman coding functions. There basic functions can be considered as behavioral clusters. Similarly, for the considered receiver’s

application, a behavioral cluster, for instance, can be a function that performs receiver's synchronization or a receiving symbol correction.

In almost all behavioral descriptions of a DECT receiver, there are behavioral clusters that their goal is to check whether an event occurs or not without modifying the output variables between the occurrences of two events. Such clusters are characterized by unobservability for one or several event windows and their shutdown can lead to significant power savings. For example, a behavioral cluster responsible for synchronization does not change its outputs for a while, after the synchronization is achieved. The granularity of a behavioral cluster complexity is user-specified. Depending on the features of an application,

the designer can specify behavioral clusters with finer or coarser granularity of complexity.

It is not always clear in an abstract behavioral description (e.g. CDFG) whether a cluster performs useful computations or not. Thus, a behavioral analysis is required to identify the clusters that can be shutdown and also the events that enable and disable these clusters. The fundamental steps of the proposed behavioral level event-driven management technique are described in Fig. 1.

4.1. Step 1: behavioral analysis

Behavioral analysis indicates the candidate clusters at the behavioral level for power management. This also involves

```

begin
Behavioral Analysis (S0, E0, C)
Protocol and/or Behavioral Analysis (tEWi,j)
extract_pm_scenaria (S0, E0, C)
characterize (E0, C)
select_optimal_pm_scenario (Si, Ei, tEWi,j,
                          Pej, aej, Pcj, acj)
end proposed_methodology

extract_pm_scenaria (S0, E0, C)
1. begin
2.   for i=1 to ||S0|| do
3.     Si = S0
4.     for all x: (ex ∉ Ei) do
5.       for all k: (∃{j,k,l} ∈ S0) ∧ (x = k) do
6.         Si = Si - {j,k,l}
7.       end do
8.       for all l: (∃{j,k,l} ∈ S0) ∧ (x = l) do
9.         Si = Si - {j,k,l}
10.      end do
11.    end do
12.    for all y: (ey ∈ Ei) do
13.      for all k: (∃{j,k,l} ∈ S0) ∧ (y = k) do
14.        if ∃ ez ∈ Ei : k < z < l
15.          Si = Si ∪ {j,z,l}
16.        end if
17.      end do
18.      for all l: (∃{j,k,l} ∈ S0) ∧ (x = l) do
19.        if ∃ ez ∈ Ei : k < z < l
20.          Si = Si ∪ {j,l,z}
21.        end if
22.      end do
23.    end do
24.  end do
25. end extract_pm_scenaria

characterize (E, C)
begin
  for j=0 to m-1 do
    compute Pcj, acj
  end do
  for j=0 to n-1
    compute Pej, aej
  end do
end characterize

select_optimal_pm_scenario (Si, Ei, tEWi,j,
                          Pej, aej, Pcj, acj)
begin
  current_min = 1
  selection = ∅
  for i=1 to ||Si|| do
    compute Costi (Eq. (10) or (11))
    if Costi < current_min
      current_min = Costi
      selection = Ei
    end if
  end do
end select_optimal_pm_scenario

```

Fig. 1. Algorithmic description of the proposed power management technique.

the identification of the events that can trigger the shutdown of the behavioral clusters.

Definition 1. We define as events' set the ordered set $E_0 = \{e_i | i=0,1,\dots, n-1, n \in \mathbb{N}\}$, where e_i is an event that either introduces or ceases unobservability for a certain behavioral cluster, and $n = \|E_0\|$ is the number of such events ($\|\bullet\|$ denotes the cardinality of the set \bullet). The set E_0 is ordered according to the time occurrence of the events e_i . Using mathematical notations, the behavioral analysis aims at defining the following set:

$$S_0 = \{(j, k, l) | (c_j \in C) \wedge (e_k, e_l \in E_0) \\ \wedge (e_k \text{ introduces unobservability for } c_j) \\ \wedge e_l \text{ ceases unobservability for } c_j\} \quad (4)$$

The simplest way to perform behavioral analysis is simulation. Concerning that the design of a wireless system start by a behavioral level description, using robust and mature automated tools, for instance Matlab [12], the required behavioral analysis can be performed in an easy and accurate manner. Furthermore, in many cases the simulation is not always needed, since the behavioral analysis can also be performed manually by any designer familiar with the behavioral description of the design. In any case, the behavioral analysis can be visualized by the use of an event graph.

Definition 2. We define as event graph a two dimensional graph where the horizontal dimension represents the system period and in which the events are arranged with respect to their occurrence sequence, while the vertical dimension corresponds to the behavioral clusters.

According to Definition 2, the system period is divided into several event windows. Thus, if a behavioral cluster is observable in an event window, then a line is drawn in the corresponding window. In contrary, the absence of a line in an event window means that during this event window the corresponding behavioral cluster is unobservable.

It is mentioned here that, the event graph can be used in any level of the available granularity. For example, a behavioral cluster can be a simple operation (arithmetical, logical, etc.) or a set of operations. However, the usage of fine-grain behavioral clusters can result in a large exploration space, which is difficult to be manually or even automatically managed. On the other extreme, some power management opportunities may be hidden if large coarse-grain behavioral clusters are used. Usually though, the original descriptions of the receiver algorithms are inherently partitioned into behavioral clusters based on their functionality, and it seems that this partitioning is in most cases convenient for the purpose of power management exploration.

4.2. Step 2: power management scenarios extraction

After the construction of the event graph, the alternative power management scenarios should be identified. A power management scenario corresponds to the usage of a subset E_i of the events' set E_0 , i.e. $E_i \subseteq E_0$. Specifically, every possible combination of the events that does not violate their initial ordering corresponds to a different power management scenario. If there are p events that introduce or cease unobservability for q behavioral clusters, then there are $2^p - p - 1$ different power-management scenarios, each one corresponding to the use of a different subset of events' set. Each power management scenario is characterized by different energy dissipation. A power management scenario is associated with a set S_i , $i = 1, 2, \dots, 2^p - p - 1$.

$$S_i = \{(j, k, l) : (c_j \in C) \wedge (e_k, e_l \in E_i) \\ \wedge (e_k \text{ introduces unobservability for } c_j) \\ \wedge e_l \text{ ceases unobservability for } c_j\} \quad (5)$$

Given the set S_0 , Eq. (4), a set S_i can be extracted using the routine *extract_pm_scenaria* shown in Fig. 1. Initially, after the behavioral analysis, we obtain the set S_0 as it defined by Eq. (4). Lines 4–11 describe the excluding of scenarios related with events that are not included in set E_i , which contains the under consideration events. In more details, we exclude the scenarios by the introduced unobservability (Lines 5–7) or ceased unobservability (Lines 8–10) for a cluster c_j due to an event e_x , which is not included in set E_i . For each considered event, e_y , we generated additional scenarios related either with event-signals e_z , which are generated after e_y that introduce unobservability for a cluster c_j (Lines 13–16), or with events e_z , which are generated before e_y that ceases unobservability for c_j (Lines 18–23). In more details (Lines 13–16), consider a cluster c_j and an event-signal e_y that introduced unobservability for c_j . That means that we can shut-down c_j for the time interval $T_{EW_{y,j}}$. Mention that e_y introduces unobservability and e_l ceases observability for the cluster c_j . However, if there are additional events e_z within this time interval, then we have the opportunity to shut-down c_j in time intervals $T_{EW_{z,l}}$.

4.3. Step 3: cost function evaluation

Each power-management scenario can be power-efficient or inefficient, depending on the power consumed for the generation of the *event variables* and the power saved by the shutdown of the behavioral clusters. In order to decide whether a power-management scenario is efficient or not, the first step is to generate the logic that produces the event variables and insert the generated logic in the behavioral description of the algorithm. The generation of the event variables is not always needed, since a variable that has identical behavior with an event variable may already exist

in the original behavioral description. In such cases, it is assumed that no power overhead is introduced.

Whether a power management scenario can lead to energy savings or not depends on the relation among the two following factors: (i) the amount of saved energy, which is determined by the time fraction in which the behavioral clusters are shut down and the corresponding power consumption, and (ii) the amount of energy that is spend for the additional logic, needed for the event signals generation.

The latter factor is determined by the time fraction, during which the extra logic is needed to operate, and the additional logic power consumption. Specifically, a power management scenario is energy-efficient, *if the energy consumed by the transformed behavioral description* (i.e. original + power management-related logic), E_{transf_i} , is less than the energy consumed by the original behavioral description, E_{orig} . Using mathematical notation, it is obtained that:

$$E_{transf_i} < E_{orig} \quad (6)$$

$$E_{orig} = \sum_{\forall j, c_j \in C} P_{c_j} t_{EW_{0,n-1}} \quad (7)$$

$$E_{transf_i} = \sum_{\forall j, c_j \in C} P_{c_j} t_{EW_{0,n-1}} + \sum_{\forall j, e_j \in E_0} P_{e_j} t_{EW_{0,n-1}} - \sum_{\forall j, k, l, (j, k, l) \in S_i} (P_{c_j} t_{EW_{k,l}}) \quad (8)$$

where $t_{EW_{0,n-1}} = T_{SYSTEM}$ and $t_{EW_{k,l}}$ are the duration of system period and event-window $EW_{k,l}$ respectively, while P_{c_j} and P_{e_j} are the power consumption of cluster, c_j , and the power dissipation of the additional circuit for the generation of the event variable, e_j , respectively. Hence, using Eqs. (7) and (8), Eq. (6) can be written as:

$$\sum_{\forall j, e_j \in E_0} P_{e_j} t_{EW_{0,n-1}} < \sum_{\forall j, k, l, (j, k, l) \in S_i} (P_{c_j} t_{EW_{k,l}}) \quad (9)$$

From Eq. (9), we infer that power savings can be achieved if and only if the power consumed by the additional circuit (for the event variables generation) multiplied by the duration of system period is less than the summation of the power saved by shutting down the behavioral clusters multiplied by the respective event-window duration. In other words, Eq. (9) describes the sufficient and necessary condition for a power-management scenario to be valid (i.e. to be power efficient). A cost function that assists the designer to choose among all valid power management scenarios can be derived directly from Eq. (9):

$$Energy_Cost_i = \frac{\sum_{\forall j, e_j \in E_i} P_{e_j} t_{EW_{0,n-1}}}{\sum_{\forall j, k, l, (j, k, l) \in S_i} (P_{c_j} t_{EW_{k,l}})} \quad (10)$$

Unfortunately, both event window and system period duration are not *a priori* specified because timing is fixed

at a lower level of the design flow, during scheduling. But even if scheduling is performed, events do not always occur at specific time instance. Thus, in the general case, the evaluation of the cost function of Eq. (10) is not feasible at this level, and can be in some cases performed at the RT-level after the behavioral synthesis of all the alternative designs. It is feasible though to evaluate Eq. (10) at the behavioral level in cases where time related information is part of the specifications. Then, the cost function should be evaluated for the worst case for each alternative power management scenario.

More specifically, the worst case for the cost function of Eq. (10) is the one where the shorter possible duration for the event windows and the longest duration of the system period are considered. When the duration of either an event window or the system period does not have an upper bound, then their average duration can be fed to Eq. (10). In the receiver context such information is supplied by the telecommunication protocol, which is always part of the specifications.

In many cases an area overhead is paid due to the insertion of logic for the generation of the event signals. However, the usage of Eq. (10) implies that regardless of the amount of this overhead the choice is always to trade area for power. If this is not the case, then the following alternative cost function, which also takes into account area, should be used:

$$Cost_i = \frac{\sum_{\forall j, e_j \in E_i} P_{e_j} t_{EW_{0,n-1}}}{\sum_{\forall j, k, l, (j, k, l) \in S_i} (P_{c_j} t_{EW_{k,l}})} + \gamma \frac{\sum_{\forall j, e_j \in E_i} a_{e_j}}{\sum_{\forall j, c_j \in C} a_{c_j}} \quad (11)$$

where a_{e_j} is the area occupation of the logic required for detecting the events e_j , a_{c_j} is the area occupation of behavioral cluster c_j , and γ is the weighting factor.

The proposed technique for behavioral-level power management exploration, which is summarized in Fig. 1, may be proven to be useless, if the decisions made are not passed as constraints to the next levels of the design flow. For example, if the behavioral synthesis allocates an operation contained in a behavioral cluster that was decided to shutdown for a certain time fraction, and an operation outside this behavioral cluster is allocated to the same resource, then the power management for this resource can be disabled. A behavioral synthesis algorithm that targets functional pipelined architectures and that takes into account the power-management decisions made by the proposed methodology has been developed [9]. Analysis of this algorithm is out of the scope of this paper.

5. Description of the DECT digital receiver

The proposed technique has been applied in the design and implementation of a DECT baseband receiver. The developed receiver is based on the direct conversion architecture and demodulates Gaussian Minimum Shift-Keying (GMSK)

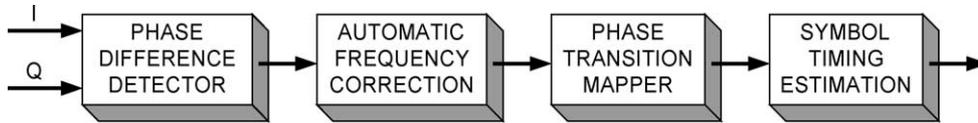


Fig. 2. The block diagram of the demonstrator application.

signals [13–15]. This approach is considered one of the most efficient since it minimizes power consumption and reduces the front-end complexity [15]. According to this technique, the intermediate frequency (IF) of down-converting stages is eliminated. This favors digital detection schemes based on in-phase (*I*) and quadrature (*Q*) channel demodulation. As a result, an all-digital implementation of the baseband receiver is allowed.

The receiver’s behavior is described by the block diagram and the corresponding CDFG are shown in Figs. 2 and 3, respectively.

In more detail, the digital DECT receiver consists of four blocks, as shown in Fig. 2. The phase difference detector (PDD) uses the In-phase (*I*) and Quadrature (*Q*) components of the received baseband signal to calculate the phase difference between two consecutive symbols. The automatic frequency correction block (AFC) uses a feed-forward technique to compensate for the frequency drifts between the local oscillators of the transmitter and receiver. The symbol decoder block (SDB) translates the corrected phase difference to a positive, negative, or zero transition which through a Finite State Machine makes a decision for the transmitted symbol. The Slot Synchronization and Symbol Timing Estimation Block (STE) is used to achieve slot synchronization and proper timing to sample the signal at the best possible instance. Further information for each receiver’s block follows. The proposed system accepts at its input a quantized, $4\times$ oversampled, IQ stream consisting of a pair (*I*, *Q*) of six-bit vectors in sign-magnitude form received on each clock cycle. The processing of the above stream yields the bit stream of the data section contained in a DECT slot, on the circuit output. Every DECT slot has

a 32-bit header, with a 16-bit preamble and a 16-bit fixed sync word, followed by the data section of 392 bits and a guard space of 56 bits.

The phase difference detector (PDD) calculates the phase difference between two consecutive symbols using a modified arc tangent function. The phase difference is represented with a fixed-point, eight-bit word in two’s complement format, having values in the range $[-\pi, \pi]$. Techniques for reducing the Look-Up Table implementation were applied and the final LUT size is $((2^5) \times (2^5))/2 - (2^5) \times 5 = 2400$ bits.

Since the receiver uses incoherent detection any frequency drift between the local oscillator of the receiver and the local oscillator of the transmitter causes a phase rotation. The automatic frequency correction (AFC) subsystem estimates the phase rotation and corrects the phase difference in a feed forward manner. An estimate for the phase rotation caused from the local oscillator frequency drift is:

$$\hat{\phi}_r(m) = \frac{1}{4} \arctan \frac{\sum_{n=0}^m \sin(4\Delta\phi(t - mT))}{\sum_{n=0}^m \cos(4\Delta\phi(t - mT))}, \quad 0 \leq m \leq M - 1$$

where *T* is the symbol period and *M* equals the number of the bits in the slot. The implementation of AFC includes two main functions: (i) the sin (cos) accumulation and (ii) the arctan function. The accumulations start on the circuit reset or when the circuit is searching for a new DECT slot. The duration of the accumulations can be at most 2.5 times the duration of a DECT slot dictated by the DECT standard. As the accumulations take place, the sine and cosine sums may increase or decrease in significant bits. However, a constant

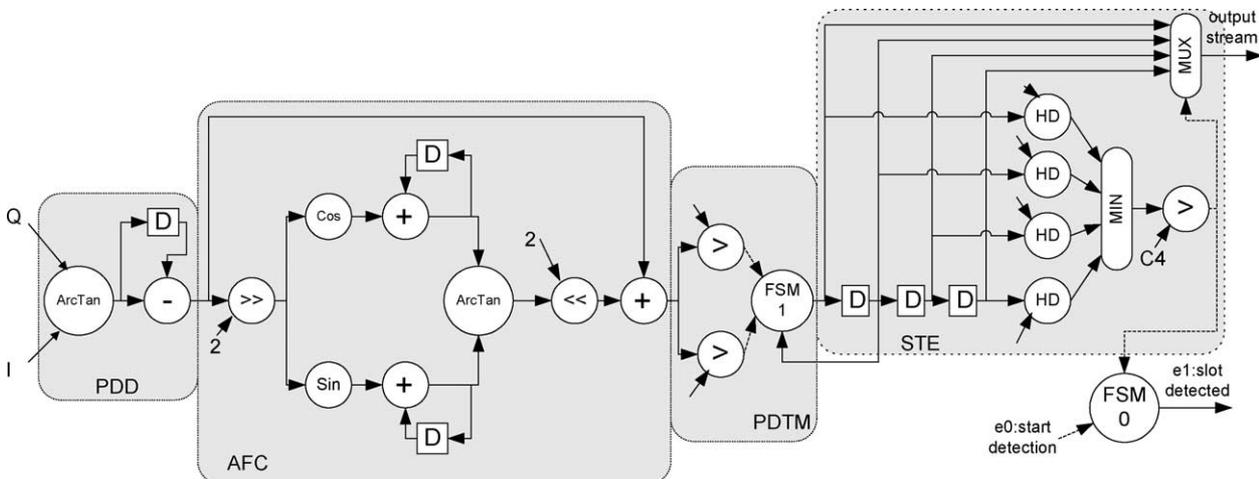


Fig. 3. The original CDFG of the DECT baseband receiver.

number of bits is needed for each operand of the arctan function. A circuit observes these sums and, in every clock cycle gives the position of the most significant bit from both sums. This serves as a select input to a multiplexer that inputs the proper five most significant bits to the arctan, along with their respective signs. The result of the arctan is divided by four and added to the original phase difference angle.

PDTM circuit consists of a comparator and a Finite State Machine (FSM). This small circuit translates the corrected phase difference (received by the AFC) to a positive, zero, or negative transition. This decision is based on a comparison of the received phase difference to two bounds ($\pi/4$ and $-\pi/4$) and selects the correct transition according to the respective area that the phase difference belongs to, as follows: (i) $(\pi, \pi/4)$: positive transition, (ii) $[\pi/4, -\pi/4]$: zero transition, and (iii) $(-\pi/4, -\pi)$: negative transition. Using the transition information and the previous detected symbol a decision on the current symbol is made by the FSM.

The DECT standard specifies each data packet starts with a synchronization field, which should be used for clock and packet synchronization of the radio link [3]. Due to lack of synchronization between the transmitter and receiver clocks, I and Q streams are oversampled by a factor of 4, in order to eliminate the possibility of sampling between symbols. The receiver uses correlation between the fixed synchronization field and samples of the four estimated sequences spaced T seconds apart to achieve synchronization. This is an attractive approach due to its inherent simplicity. The functionality of the STE block can be outlined by two steps: (i) in each of the four time multiplexed bit streams (due to oversampling), which derive from the four corresponding IQ streams, detect the preamble and sync word (the start of a DECT slot), allowing a user-defined maximal number of errors and (ii) among the bit streams that have match the above criterion, select the one with the smallest number of errors. In order for a bit stream to be eligible for selection, the 16-bit sync word should be detected with no more than T1 errors, and the last four bits of the preamble should be detected with no more than T2 errors. Once these restrictions are satisfied for one bit stream, the circuit stores the number of total errors in the whole 32-bit header. If in the next three clock cycles another bit stream satisfies these restrictions with a lower total error count, which bit stream will be selected as the optimum. This bit stream is most probably sampled at the closest to optimal time point, thus correct symbol timing estimation is achieved along with correct slot detection.

6. Proposed technique applied in the DECT receiver design

The behavior of the receiver is as follows: The first block calculates the phase difference of two consecutive samples

(phase difference detector—PDD cluster). The phase differences are estimated and corrected by the automatic frequency correction (AFC) cluster. The phase difference transition mapper (PDTM) cluster decodes corrected phase difference and the transmitted sequence is recovered. Additionally, the Symbol Timing Estimation (STE) cluster is responsible for slot synchronization and symbol timing estimation. A more detailed description as well as the design and implementation of the DECT receiver can be found in [2].

For the purpose of power management exploration, the behavioral clusters taken under consideration are: $c_0 \rightarrow$ STE, $c_1 \rightarrow$ PDTM, $c_2 \rightarrow$ AFC, and $c_3 \rightarrow$ PDD. The behavioral analysis indicates that c_0 does not need to operate after the header (synchronization and preamble) detection and up to receiving the whole slot. Also, c_2 computes a series in order to estimate the error due to the frequency drift, which ones converges, does not modify its output for the rest of the slot. The corresponding event graph is shown in Fig. 4.

The events that divide the system period and specify the event windows are: (i) the start detection (e_0), (ii) the slot detection/synchronization (e_1), (iii) the error convergence (e_2) and (iv) the end of slot (e_3). The event variable e_0 and e_1 are already present in the original CDFG (Fig. 2). The additional logic for the generation of the event variables e_2 and e_3 is shown in the transformed CDFG of Fig. 5.

In the worst case (minimum duration) of the event window $EW_{2,3}$ is equal to the three-fifths of the duration of a slot. Also, the duration of $EW_{1,3}$ is always equal to the duration of a slot. The worst case (maximum duration) for system period ($EW_{0,3}$) is not fragmented but according to a probabilistic approach, the system period is on average 2.3 times the duration of the slot.

From the eleven candidate power management scenarios, only 3 succeed to shutdown at least one behavioral cluster for an event window; for the rest 8 scenarios the value of the cost function (Eq. (10)) is greater than one (i.e. inefficient scenarios). The first scenario uses all the event variables. The second one uses e_0 , e_1 and e_3 , while the third one uses the variables e_0 , e_2 and e_3 . In the case that all the event variables are used, then c_0 can be shutdown for

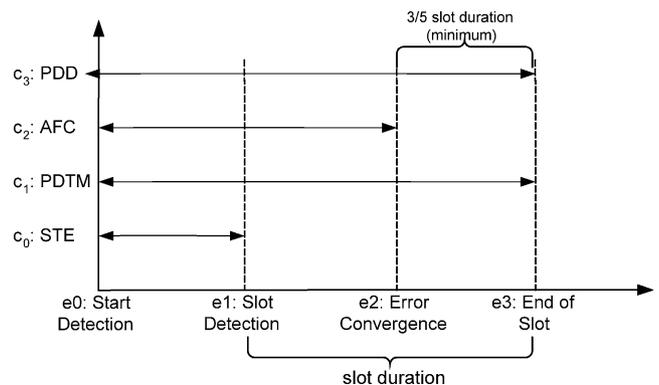


Fig. 4. The event graph for the DECT baseband receiver.

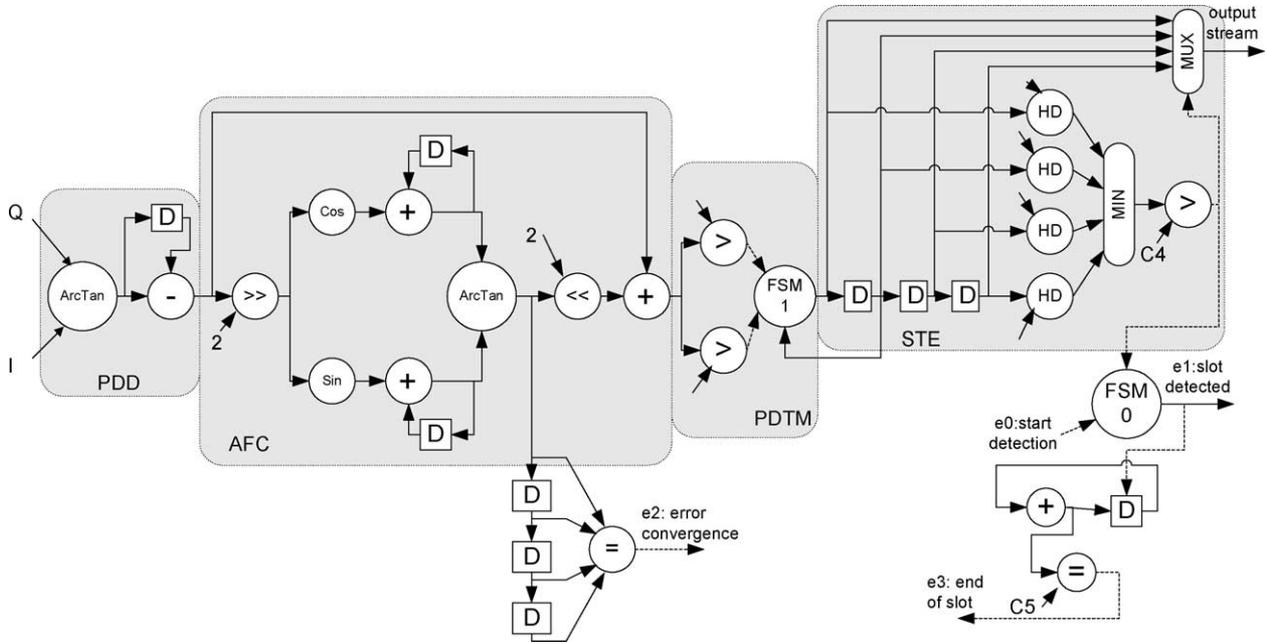


Fig. 5. The transformed CDFG of the DECT baseband receiver.

the event-window $EW_{1,3}$, and c_2 can be shutdown for $EW_{2,3}$ ($S_0 = (\{0,1,3\}, \{2,2,3\})$) and additional logic is required for the generation of e_1 and e_2 . In the case that e_0 , e_1 and e_3 are used then the only behavioral cluster that can be shutdown is c_0 ($S_1 = (\{0,1,3\})$) and obviously only the additional logic that produces e_1 is needed. In the case that e_0 , e_2 and e_3 are used, c_0 and c_2 can be shutdown during the event window $EW_{2,3}$ ($S_2 = (\{0,2,3\}, \{2,2,3\})$) and the logic that produces the event variables e_2 and e_3 is needed.

Using the cost function of Eq. (10) it was inferred that the most efficient power-management scenario is S_1 . Specifically, the valid power management scenarios S_0 , S_1 , and S_2 , achieve power saving of 10, 45, and 20%, respectively. This means that the presence of the logic that generates e_2 introduces energy overhead greater than the energy saved by shutting down the AFC block during the event window $EW_{2,3}$. This is due to the relatively small duration of $EW_{2,3}$.

The cost function values and energy measurements acquired after circuit implementation, which corresponds to each one of the valid power-management scenarios, and logic-level simulation are illustrated in Fig. 6. The measurements denoted as *Cost Function* in Fig. 6 have been derived using the procedure described in Sections 2 and 3 (step 3). The modified DECT receiver circuit has been implemented in Matlab and a functional simulation has been performed to find out the Hamming distances at the input/output nodes of each basic operation. Afterwards, using the capacitance models of [11] the power consumption, P_{c_j} , of each behavioral cluster c_j have been evaluated. The values of P_{c_j} have been also evaluated in the same manner. Finally, using Eq. (10) the cost function for possible power management scenario has been computed.

Regarding the measurements denoted as *Measure* in Fig. 6 they are derived by describing the modified circuits, which are correspond to the candidates power management scenarios, in VHDL, synthesizing them and perform logic-level simulation. Thus, the number of transitions of each circuit node has been derived and using the extracted capacitance values of the synthesized circuits the dynamic power dissipation was evaluated. Comparing the estimated power consumption values derived by using the proposed power model and those values derived after circuit implementation and simulation, it is clear that the used power model evaluates the dissipated power with an accuracy that is adequate concerning the design level (i.e. behavioral level) where the proposed technique is applied. It is mentioned that S_1 , which is the most power efficient scenario, introduces an area penalty less than 5% and do not increase the delay of the critical path.

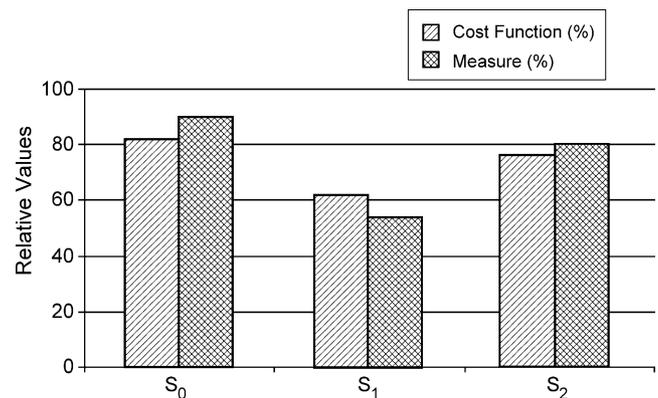


Fig. 6. The cost function evaluation.

7. Conclusions

A new behavioral-level event-driven power management technique applicable in the digital receiver context was proposed. This approach targets the exploration of the trade-off according to which on the one hand power is saved by shutting-down parts of the circuit but on the other hand power is increased by the additional logic required. The application of the proposed technique in the design of a real-life DECT baseband demodulator has proven that significant energy savings can be achieved.

References

- [1] ETSI, DECT Specification Part 2: Physical Layer, ETS 300 175-2, July 1995.
- [2] N.D. Zervas, et al., Low-power design of direct conversion baseband DECT receiver, *IEEE Transactions On Circuit and Systems—Part II* 48 (2001) 1121–1131.
- [3] J. Rabaey, M. Pedram, *Low Power Design Methodologies*, Kluwer, Dordrecht, 1995.
- [4] L. Benini, G. de Micheli, *Dynamic Power Management: Design Techniques and CAD tools*, Kluwer, Dordrecht, 1998.
- [5] L. Benini, G. de Micheli, E. Macii, M. Poncino, R. Scarsi, Symbolic Synthesis of Clock-Gating Logic for Power Optimization of Control-Oriented Synchronous Networks, *Proceedings of European Design and Test Conference*, Paris, France 1997; 514–520.
- [6] M. Aldina, J.S.N. Monteiro, A. Ghosh, M. Papaefthymiou, Pre-computation-based sequential logic optimization for low power, *IEEE Transactions on VLSI Systems* 2 (4) (1994) 426–436.
- [7] V. Tiwari, S. Malik, P. Ashar, Guarded Evaluation: Pushing Power Management in Logic Synthesis/Design *Proceedings of the International Symposium on Low Power Design*, Dana-Point, CA (1995), pp. 221–226.
- [8] L. Benini, A. Bogliolo, G. de Micheli, A survey of design techniques for system-level dynamic power management, *IEEE Transactions on VLSI Systems* 3 (3) (2000) 299–316.
- [9] N.D. Zervas, D. Soudris, C.E. Goutis, A. Thanailakis, *Low-Power Methodology for Transformations of Wireless Communications Algorithms*, Deliverable report LPGD/WP2/DUTH/D2.2R1, 1999.
- [10] C. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [11] Landman P., *Low power architectural design methodologies*, PhD Dissertation, UC Berkeley, August 1994.
- [12] Matlab-Simulink: <http://www.mathworks.com>
- [13] J.G. Proakis, *Digital Communications*, Mc Graw-Hill, New York, 1995.
- [14] J.D. Gibson, *The Communications Handbook*, CRC Press, Boca Raton, FL, 1997.
- [15] G. Schultes, P. Kreuzgruber, A.L. Scholtz, DECT Transceiver Architectures: Superheterodyne or Direct Conversion?, *Proceedings of the IEEE 43rd Vehicular Technology Conference*, May 1993, pp. 953–956.