

A LOW POWER SIGNAL DETECTION AND PRE-SYNCHRONIZATION ENGINE FOR ENERGY-AWARE SOFTWARE DEFINED RADIO

Bruno Bougard (IMEC, Leuven, Belgium; bougardb@imec.be); Lieven Hollevoet (IMEC, Leuven, Belgium, hollevo@imec.be); Frederik Naessens (IMEC, Leuven, Belgium, naessens@imec.be), Thomas Schuster (IMEC, Leuven, Belgium, schuster@imec.be); Ching Ho (Anthony) Ng (IMEC, Leuven, Belgium, nga@imec.be); Liesbet Van der Perre (IMEC, Leuven, Belgium; vdperre@imec.be)

ABSTRACT

SDR enables cost-effective multi-mode terminals but still suffers from significant energy penalty when compared to dedicated hardware solutions. At system level, this energy bottleneck can be leveraged by capitalizing on the opportunistic partitioning and energy-scalable design of both hardware and software architectures. This yields MPSOC platforms where specific engines are dedicated to classes of functions that relate in their computation characteristics and in their duty cycle. In case of burst-based signal reception, detection functions have high duty cycle and hence need ultra low power implementation. Besides, signal synchronization as close as possible to the ADC is desired to free the system bus of signal-less data samples, with direct impact on the system performance and energy. A specific, still programmable ultra low power detection and pre-synchronization engine targeted to IEEE802.11a/g/n and IEEE802.16e signals is designed in 90nm CMOS. Results show that detectability is guaranteed with a minimal standby power of 1.1 mW, valid signal detection and pre-synchronization consumes 228nJ while false trigger by a blocker account for 300nJ.

1. INTRODUCTION

The combination of the continuously growing variety of wireless standards and the increasing cost related to IC design and handset integration make implementation of wireless standards on reconfigurable radio platforms the only viable option in the near future. An effective solution is the combination of reconfigurable analog front-end circuits and SDR-based digital baseband platforms. However, to be viable in the handhelds market, a reconfigurable radio platform must be *low cost* (both silicon and NRE costs must be contended) and *low power*.

If programmable in a high-level language (such as C), SDR enables cost-effective multi-mode terminals but still suffers from a significant energy penalty when compared to dedicated hardware solutions. Hence, one has to carefully trade off programmability and energy efficiency. To maintain energy efficiency at the level required for mobile device integration, programmability may only be introduced where its impact on the total average power is sufficiently low or at those places where the resulting extra

flexibility can be exploited to yield an average energy gain through better matching of the system behavior to the utilization and the environment (energy-scalable design, [1]). Many different architecture styles have already been proposed for SDR. Most of these are designed keeping in mind the most important characteristics of wireless physical layer processing: *high data level parallelism* (DLP) and *data flow dominance*. For the first characteristic, hybrid VLIW and vector/SIMD architectures are often considered to exploit the data level parallelism with limited instruction fetching overhead [2,3,4]. However, directly mapping C-code, even with high DLP, on such architectures remains a challenge for the compiler. The second characteristic is exploited by fine-grain reconfigurable arrays (FGA) [5,6] and coarse grain reconfigurable arrays (CGA) [7,8]. The main bottleneck of the FGAs is the high interconnect cost that hampers their scalability and that yields significant energy overhead. CGAs improve on this point proposing less but more complex functional units. The main challenge of CGAs for SDR – their programmability in high-level languages – is addressed in [9]. Combining aforementioned architecture paradigms, one can trade off energy efficiency, performance and software development productivity. A hybrid CGA-SIMD architecture is proposed in [10], based on the core technology described in [9]. However, none of the listed architecture frameworks provides the “targeted flexibility” required for energy efficiency. This is mainly due to the fact that only the characteristics of the modulation/demodulation baseband processing are considered. In practice, a radio standard implementation also contains functionalities for medium access control and, in case of burst-based communication, signal detection and time synchronization. The data level parallelism characteristic does not hold for medium access control (MAC) processing which is, by definition, control dominated and, hence, better fit on RISCs. Besides, packet detection and coarse time synchronization of burst-based transmission have a significantly higher duty cycle than packet modulation and demodulation. They hence require another flexibility/efficiency tradeoff.

Targeted flexibility actually claims for heterogeneous MPSOC architectures. Such architectures are explored in [3,11]. In this work, we take a quite similar approach but

go further in terms of targeted flexibility and energy scalability by developing specific programmable engines for ultra low power burst detection and pre-synchronization. The proposed design is integrated in our SDR platform and characterized in terms of energy consumption by measurement in two specific scenarios: reception of a valid burst, and triggering by an invalid signal. Based on those results, the average energy consumption is extrapolated for a range of utilization conditions.

The remainder of this paper is structured as follows. In section 2, the considered MPSOC SDR platform template is described. Section 3 focuses on the detection and pre-synchronization engine architecture and its benchmarking. Measured and simulated performance and power consumption results are presented in Section 4. Conclusions are finally drawn in Section 5.

2. SDR PLATFORM TEMPLATE

The considered platform template is depicted in Figure 1. It is specifically designed to support SDR implementation of IEEE 802.11n WLAN (assuming both multi-antenna space-division-multiplexing (SDM) and channel bonding) and IEEE 802.16e mobile wireless broadband access. Provisions are taken to enable future SDR implementation of the emerging 3GPP-LTE broadband cellular standard.

As in [4], a RISC platform controller (ARM926EJ-S in our prototype, [12]) is responsible for the MAC functionality and the PHY processing macro-pipeline scheduling. This core is coupled through a multi-layer AHB bus to three types of processing units: digital front-end (DFE) tiles, baseband modem engines and forward error correction (FEC) engines.

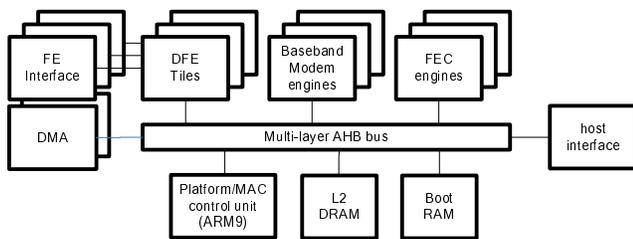


Figure 1 Top level view of the SDR platform template

A digital front-end tile implements functions relative to analog front-end control and I/Q sample interfaces, analog front-end steering (e.g., automatic gain control), signal detection, decimation, burst pre-synchronization (in receive mode) and signal interpolation (in transmit mode). To support multi-antenna operations, three DFE tiles are coupled with three analog front-end signal paths. These however share a single bus interface. FIFO buffering (32kbit) enables burst transfer of the samples that will be transmitted or of the pre-synchronized received samples over the SoC interconnect and to ease power management at platform level allowing a longer start-up time.

The baseband processing units, on the other end, implement receive functions relative to fine synchronization, front-end impairment compensation, multi-antenna processing (e.g., SDM equalization) and demodulation (OFDM). In transmit mode, channel encoding and modulation are implemented on the baseband units too. The considered processor architecture is based on the hybrid SIMD-CGA approach presented in [10].

Forward error correction engines accelerate the data decoding from the demodulated streams. Only Viterbi decoders are implemented in the current prototype although the MPSoC template can easily be upgraded with turbo- or LDPC accelerators.

Each class of processing units has a different programmability/performance/energy-efficiency tradeoff. For the baseband processors, programming productivity and performance are the main concerns. Energy-efficiency may be slightly relaxed because of the lower duty cycle and the possible energy-scalable implementation of the supported functions. FEC requirements are more homogeneous over the standards so that parametrical VLSI implementations are still an efficient option. Digital front-end units however require a more balanced tradeoff between programmability (to be able to detect burst from different standards) and energy efficiency (as they are almost continuously active and, hence, are the main contributors to the standby power).

Such opportunistic partitioning, when combined with aggressive power management, is extremely valuable when implementing burst-based communication standards.

3. BURST DETECTION AND PRE-SYNC ENGINE

Programmable solutions intrinsically suffer from higher power consumption when compared to dedicated solutions. In our platform, the digital front-end tiles, which implement signal detection and pre-synchronization functions, require both very high energy efficiency and sufficient programmability to implement detection of different standards on the same tile. The key to this combination lies in the hierarchical power up of the DFE functionality. An architecture that enables such hierarchical wakeup is proposed in this section.

3.1 Top level architecture

The DFE consists of multiple ‘tiles’. A single tile contains the digital receive and transmit logic to interface to a single antenna.

The transmitter part of a DFE tile consists of a buffer and a VLSI interpolation filter. The interpolation filter is based on an optimized implementation of a 19-taps half-band filter (with hamming window) for a fixed upsampling of factor two.

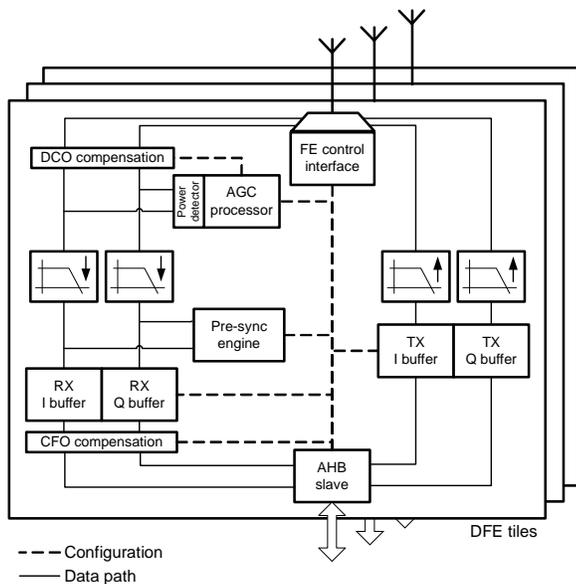


Figure 2 Digital Front End architecture

A start command can be issued allowing the samples to be clocked out towards the analog front-end through the filters. The transmit (TX) buffers have a programmable threshold that triggers an interrupt once the number of available samples falls below this threshold. This interrupt is handled by the platform controller.

The receiver part of a DFE tile contains a chain made of the VLSI decimation filters, the buffers and compensation units for DC offset and carrier frequency offset (CFO). The decimation filter impulse response is derived from a 19-taps half-band filter with hamming window performing an energy efficient factor two downsampling. Next to the datapath, two dedicated micro-processor cores are implemented. The first handles the front-end automatic gain control (AGC) and the DFE power management. The second core is optimized for time synchronization.

3.2 Programmable power detection and AGC controller

A power detection unit with variable delay line of 8 or 32 samples determines the received signal power and is capable of DC offset estimation. A dedicated microcontroller is used to implement the AGC algorithm that removes DC offset and optimizes the ADC range based on the analog front-end control pins. The controller also determines which other parts of the DFE RX are activated after an AGC event is detected (gradual wakeup). The controller architecture is depicted in Figure 3, it is clocked at the sample rate (40MHz). It has an instruction memory of 512 14-bit words and a data scratchpad of 32 8-bit words, both implemented as registerfile macros. The instruction set and the architecture of the controller are compatible with an industry-standard Microchip PIC16F84 [13]. This allows for reuse of the available toolchains, including c-compilers and debuggers.

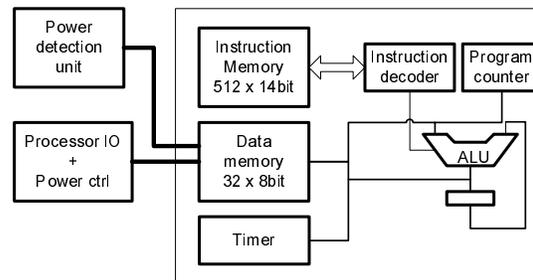


Figure 3 AGC controller architecture

3.3 Signal detection and pre-synchronization processor

As soon as the AGC controller detects the presence of a potential signal and has optimized the ADC range accordingly, a dedicated application specific processor (ASIP) is activated to perform coarse time synchronization. The ASIP has a 2-issue VLIW architecture (Figure 4). The first slot is made of a 16-bit data path for address computation and control. The second slot is a 128 bit vector unit. The vector unit is dominated by a complex vector multiplier, which can operate on four complex samples in parallel. The processor also contains a vector scratchpad of 32kbit (256x128bit) and an instruction memory of 20kbit (512x40bit). The instruction set is depicted in Table 1. Based on that IS, approximately 300 instructions are sufficient to implement both 802.11a and 802.16e synchronization loops. Although, the 802.16e synchronization is far more computing intensive, the processing steps are similar for both modes.

The synchronization algorithm is executed as follows. A data vector is fetched from the ASIP scratchpad, which always contains a copy of the main data path FIFO content. Next, the correlation and the input signal power are calculated. In the case of the 16e mode, this includes keeping track of a set of moving sums. However, for both modes we determine the running maximum of the correlation with respect to the input signal power. Whenever a correlation peak above a defined threshold is detected, the computation stops, the index of the maximum is written to an output port and the *sync* signal is asserted. The sync signal is then interpreted by the DFE power management processor, which in turn disables the synchronization processor and wakes up the baseband part of the platform so that the received data can be transferred for processing. For real-time 802.11a/g/n synchronization, the processor has to run at minimum 130 MHz. The 802.16e mode requires a clock rate of 280 MHz. These clocks are provided with a local PLL (independent from the SoC main clock generator) which is fed up with the sample clock divided per two. The ASIP clock is set to 140MHz or 280MHz depending of the mode it is configured to receive. In case of 140MHz operation, voltage can be reduced from 1V to 0.8V.

TABLE 1 DETECTION AND PRE-SYNC ASIP INSTRUCTION SET

Scalar unit	
nop	No operation
pinld	Load vector from external port
pinst	Store to external port
ldv	Load vector from DMEM
stv	Store vector from DMEM
mov	Move registers
add	Add registers
sub	Subtract registers
mul	Multiply registers (16bit)
bneg	Branch on negative value
rmax	Maximum of real parts in vector
rgrep	Extract value from real parts of vector
spread	Fill vector with complex value
Vector unit	
nop	No operation
vadd	Vector addition
vcmp	Vector comparison
vcon	Vector conjugate complex
cmul	Vector complex multiplication
triang	Vector accumulation with interm. results
real	Real parts of vector
level	Fill vector with element from vector

```

iter:
    pinld V2          || vcmp V7, V7, V6
    ldv V3, R1        || nop
    stv R0, V2        || vcon V5, V2
    ldv V4, R2        || cmul V2, V2, V5
    nop               || vadd V4, V3, V4
    add R0, R0, R15   || cmul V4, V4, V5
    add R1, R1, R15   || triang V2, V2, V1
    add R2, R2, R15   || level V1, V2
    nop               || triang V4, V4, V0
    rmax R4, V7       || level V0, V4
    rgrep R5, R4, V6  || vcon V5, V4
    spread V7, R3, R13 || cmul V6, V4, V5
    nop               || real V2, V2
    sub R6, R5, R3    || cmul V2, V2, V2
    nop               || real V7, V7
    bneg R12, R6      || cmul V7, V7, V2
    mov R3, R5        || nop
    mov R8, R4        || nop
    jmp @iter         || nop

thres:
    sub R5, R7, R3    || nop
    bneg R14, R5      || nop
    jmp @iter         || nop

sync:
    mov R12, #12     || nop
    mul R0, R2, R10   || nop
    sub R0, R0, R12   || nop
    add R0, R0, R8    || nop
    pinst R0          || nop
    jmp @init        || nop
    
```

Figure 5 ASIP code for 802.11a/g/n synchronization

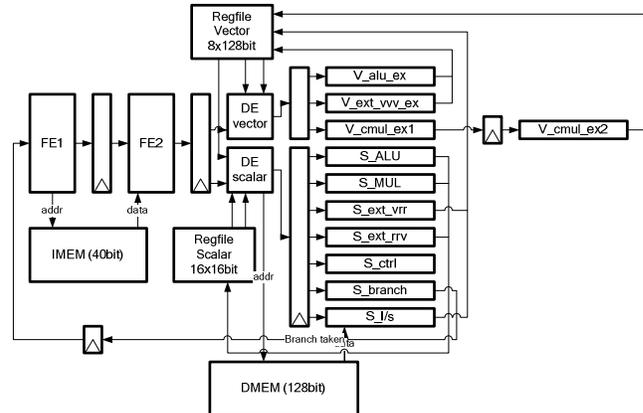


Figure 4 Signal detection and pre-sync processor architecture. Besides the traditional ALU, MUL, Ctrl, branch, L/S units, specific V_ext_vvv_ex, S_ext_vrr and S_ext_rrv take care respectively of combining 2 vectors in one, splitting one vector in scalars, combining scalars in a vector.

3.4 Top level control and power management

For data transmission, the TX buffers and interpolation filters are powered up. An ‘almost empty’ threshold is programmed in the transmit buffer block, allowing an interrupt driven loading method of the buffers. Once a start command is sent to the transmit buffers, samples are clocked out of the buffers, filtered, and sent towards the analog frontend. At the end of the burst, the transmit buffers and interpolation filters are put back in sleep mode.

A burst reception through the RX part of a DFE tile is organized as follows. When the receive part is activated, only the AGC controller is powered. The frontend is programmed with its initial settings. When the receive threshold power is reached, the controller performs the AGC algorithm, and a fine DC offset estimation is performed. The DC offset is fed back to the compensation unit. At the same time, the receive filters, the sample buffer and the synchronization ASIP are activated. Once the ASIP detects a packet preamble, the sample index at which synchronization was reached is stored and the ASIP is powered down. The synchronization point is then adapted by a programmable offset to determine the first sample of the burst that will be transferred towards the baseband processor.

The platform controller is notified of the synchronization event by the DFE microcontroller asserting a platform-level interrupt. During the wake-up time of the baseband processor, samples are stored in the DFE RX buffers. These buffers have a programmable threshold level, which again enables event driven buffer readout. At the end of the burst, the complete receive path, except for the AGC and power detection unit, is powered down and a new burst detection can take place.

In case of a blocker signal, AGC is performed but no synchronization event occurs. After a time out counted down

by the AGC controller timer, the buffers are flushed and turned off, the filters and synchronization ASIP are deactivated.

Finally, the datapath of the RX DFE is augmented with a carrier frequency offset (CFO) compensation unit that is capable of performing a rotation on the time domain signal. The settings of that block are programmable through the AHB bus interface. The CFO value is programmed to be the measured value of the previous burst and can be updated during the reception burst if required.

At the end of the burst, the complete receive path, except for the AGC and the power detection unit, is powered down and a new burst detection can take place.

4. PERFORMANCE AND POWER CONSUMPTION

All the aforementioned components have been designed and synthesized with a typical 90nm CMOS technology. Artisan standard cell and registerfile macros are considered. For SRAMs, Mosaid Mobilized™ gate-bias macros are considered. Nominal supply voltage is 1V. AGC, filter and buffer are designed with high VT cells. For the detection and pre-sync ASIP, Nominal VT is considered. When working at low speed, the ASIP can operate at VT=0.8V. but this is not considered in this paper. Synthesis is done with Synopsys Physical Compiler assuming the worst case design corner (VDD=0.9; T=215C; slow process).

In this section, we focus on the function blocks required in reception mode. In transmit mode, the contribution of the DFE can be neglected compared to the rest of the SDR SoC. The power consumption of the AGC controller with its power measurement line, its IMEM and DMEM, the decimation filter and the detection and synchronization ASIP (with IMEM and DMEM) has been evaluated using Synopsys PrimePower™. Power simulation is done at gate level for each entity separately with test vectors corresponding to the execution of their respective code for 802.11a/g/n detection, synchronization and buffering. The bus interface is not considered in our experiment. The static and dynamic power consumption figures of the different components are summarized in Table 2.

TABLE 2 DFE RX COMPONENT POWER CONSUMPTION

Component	Active (mW)	Static (μW)	Sdby (μW)
AGC datapath	.40	2	-
AGC controller	.13	.7	-
AGC IMEM	.20	2	1
AGC DMEM	.34	3	-
Decimation Filters	.86	4	-
FIFO buffer	2.74	27	-
Sync ASIP core	14.27	71	-
Sync ASIP IMEM	2.47	25	12
Sync ASIP DMEM	1.37	13	-

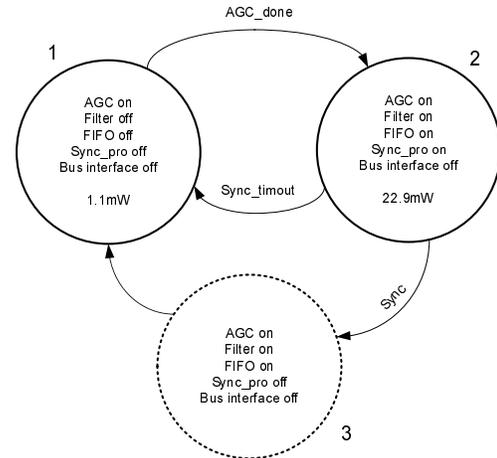


Figure 6 Power state machine

Based on this data and considering the different combination of active component together with the possible transitions, the system power state machine can be derived (Figure 6). Transition times are further investigated. In the following, the power state machine is used in order to evaluate the average power needed to guarantee detection of a burst (detectability) and the energy spent in the detection and pre-synchronization of a valid signal or the detection and reject of a blocker signal.

4.1 Minimum power consumption for burst detectability

To guarantee detectability of a potentially incoming burst, only the AGC controller and power detector datapath must be activated. The other components are completely switched off by means of power gating. However, to guarantee seamless start up, the instruction memory of the synchronization processor must be kept supplied. Using substrate-biased SRAM, as provided by Mosaid Mobilized, one can still significantly reduce the leakage. Hence, the minimum power for detectability is the sum of, on the one hand, the AGC active and static power, its memories and, on the other hand, the leakage power of the synchronization processor program memory when set in retention sleep mode. Totally, the minimum power for detectability is only 1.1 mW.

4.2 Energy for valid burst detection and pre-sync.

The sequence of operations required to guarantee the detection and pre-synchronization of a valid burst (802.11a case) is illustrated in Figure 7. A systemC model of the system is fed with sample data obtained with IMEC 802.11a receiver VLSI implementation [14]. The AGC_enable signal is high when the DFE tile is active. The AGC controller is continuously analyzing the incoming data. Power detection is signaled by AGC_done (on time index 18025ns in our simulation). This yields the assertion of the sync_enable, filter_enable and buffer_enable signals that activate respectively the syn-

chronization processor, the decimation filters and the data FIFO. The synchronization processor then executes the code illustrated in Fig. 5. For the considered input signal, a synchronization event occurs at time index 27675 (sync signal). This causes the assertion of a platform level interrupt (DFE_int), which wakes up the platform controller.

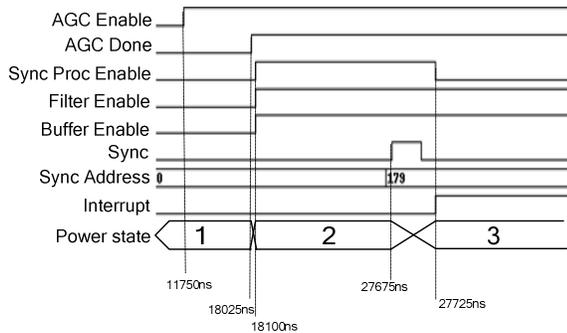


Figure 7 Activity trace when detecting a valid burst

The power state flow is appended to Figure 7. Summing up the state power multiplied by the state duration, one can easily compute the energy consumed during the burst detection. Specifically, we consider the energy spent between the reception of the first valid sample until the generation of the DFE_int interrupt. In the current experiment, this gives 228nJ.

4.3 Energy lost due to false trigger

Similarly, the sequence of operation occurring at the reception of a blocker signal (false trigger) is depicted in Figure 8. Although an AGC_done signal is generated and the filter, buffer and synchronization processor are activated, no synchronization point is found and hence, the 'sync' signal is not asserted. Filter, buffer and synchronization processor are forced back to sleep mode after a time-out occurs at time index 31025ns.

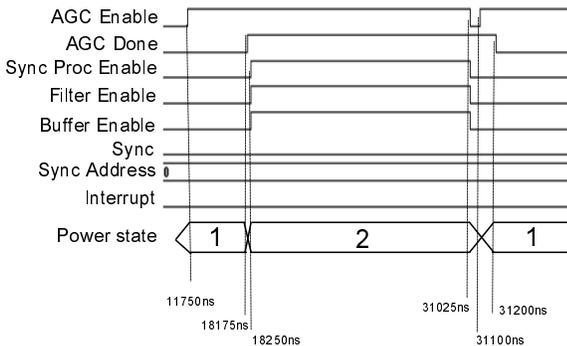


Figure 8 Activity trace when detecting a false burst

The state flow is again appended to Figure 8 and the energy spent in the false trigger event is computed similarly, giving 300nJ. The average power during the false trigger event is 15.2mW. Therefore, in field operation where false trigger occurs with probability p , the consumption of the DFE tile would hence be $1.1(1-p) + 15.2p$ mW.

5. CONCLUSIONS

SDR enables cost-effective multi-mode terminals but still suffers from significant energy penalty. Heterogeneous MPSoC resulting from opportunistic partitioning and energy-scalable design are suitable to leverage that penalty. In case of burst-based signal reception, detection functions have high duty cycle and hence need ultra low power implementation. A specific, still programmable ultra low power detection and pre-synchronization engine targeted to IEEE802.11a/g/n and IEEE802.16e signals is designed in 90nm CMOS. Results show that detectability is guaranteed with a minimal standby power of 1.1 mW, valid signal detection and pre-synchronization consumes 228nJ while false trigger by a blocker account for 300nJ. The power during the false trigger event is 15.2mW. Therefore, in field operation where false trigger occurs with probability p , the consumption of the DFE tile would hence be $1.1(1-p) + 15.2p$ mW.

REFERENCES

- [1] A. Sinha, A. Wang, and A. P. Chandrakasan, "Energy Scalable System Design," *IEEE Transactions on VLSI Systems*, Vol. 10, No. 2, pp. 135-145, April 2002, Transaction on VLSI Systems, Apr. 2002
- [2] K Van Berkel, F. Heindle, P. Meuwissen, K. Moeren and M. Weiss, "Vector Processing as an Enabler for Software-Defined Radio in Handsets from 3G+WLAN Onwards," *Proc of SDR Technical Conference*, pp. 125-130, November, 2004.
- [3] J. Glossner *et al.*, "A Software Defined Communications Baseband Design," *IEEE Communication Magazine*, Vol. 41, No. 1, pp 120-128, Jan. 2004
- [4] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, K. Flautner, "SODA: A Low Power Architecture For Software Radio," *Proc. Of ISCA*, IEEE, 2006
- [5] A. Lodi *et al.*, XiSystem: A XiRISC SoC with reconfigurable IO Module, *IEEE Journal of Solid State Circuit*, (41)1, pp 85-96, Jan 2006
- [6] G. Desoli and E. Filippi, An Outlook on the Evolution of Mobile Terminals, *CAS Magazine*, second quarter 2006.
- [7] I. Chen *et al.*, Overview of Intel's Reconfigurable Communication Architecture, *Proc. 3rd Workshop on Application Specific Processors*, pp. 95-102, Sept 2004.
- [8] N. Bagherzadeh *et al.*, MorphoSys: A Parallel Reconfigurable System, *Proceedings of Euro-Par 99*, France, Sep 99.
- [9] B. Mei, S. Vernalde, D. Verkest, H. De Man and R. Lauwereins, "DRESC: A Retargetable Compiler for Coarse-Grained Reconfigurable Architectures," *Proc of Field Programmable Technology*, pp-166-174, 2002
- [10] D. Novo *et al.*, "Mapping a multiple antenna SDM-OFDM receiver on the ADRES coarse-grained reconfigurable processor," *Proc. IEEE Workshop on Signal Processing Systems*, Athens, Nov. 2005
- [11] B. Bougard, D. Novo, F. Naessens, L. Hollevoet, T. Schuster, M. Glasse, A. Dejonghe, L. Van der Perre, "A scalable programmable baseband platform for energy-efficient reactive software-defined radio", *Crowncom 2006*
- [12] <http://www.arm.com/products/CPU/ARM926EJ-S.html>
- [13] Microchip "PIC16F84A datasheet", <http://www.microchip.com>