eSAME 2015 Conference

SMART ELECTRIC METERING FOR EFFICIENT ELECTRIC NETWORK

Trung Kien Nguyen[1,2], Eric Dekneuvel [1], Gilles Jacquemod[1], Benjamin Nicolle[3] , Olivier Zammit[3],
Cuong Nguyen Van[2],

[1]EpOC, URE UNS 006, Sophia Antipolis, France
[2]Danang University of Science and Technology, Vietnam
[3]Qualisteo Company, Nice, France

Abstract:

Non-Intrusive Appliance Load Monitoring (NIALM) meter is an innovative technology that uses only one sensor node to monitor the power consumption of individual electric appliances in an electrical network. SoC technology can solve the challenge of a compact, low cost and low power consumption NIALM smart meter. However, a NIALM system contains both two processing types: data streaming process to monitor power consumption in appliances and the event process to respond to consumer's activities. Thus, this NIALM meter is an instance of Reactive Process Network (RPN) system with hard constraints in timing, memory usage and power consumption which requires an efficient development approach. In this paper, we present applying a new SoC development approach to develop a NIALM system with the ability to quickly evaluate complex RPN system in real prototype on SoC. In this approach, synchronous dataflow (SDF) model and StateChart are used to develop the system quickly from the application model to prototype.

Keywords: NIALM, Synchronous Dataflow (SDF), StateChart, FPGA, SoC.

I. INTRODUCTION

People so far are using coal-fired to generate 40% global electric power. This energy resource emits over 40% CO2 to environment [1] that may cause the global warming and natural disasters such as drought, flooding, tsunami, hurricane, diseases to human life. Consequently, changing people's awareness of using energy effectively and economically becomes very urgent. Thus, smart meters - the emerging technology to help people to know their energy consumption, are gradually replacing mechanical power meters. The new modern NIALM smart meters even offer real-time feedback on the in-home display with detailed daily records of activities of consumer in appliances and their effect on the total energy consumption as well as the ratio of energy usage in individual appliance.

As illustrated in fig.1, in comparison to conventional meters, NIALM meters require added-value algorithms to analyse, classify, breakdown and assign power consumption to the appropriate appliances. In order to do that using only one sensing node, NIALM systems must have five basic processes.

- Data acquisition to collect data from current and voltage sensors;
- Pre-processing to extract necessary electrical information such as power, harmonic etc.;
- Event Detection to detect changes in electrical signatures relating to operation of appliances;
- Classification to classify appliances from processed electrical data;
- Estimation to summarize ratios of power consumption in each appliance.

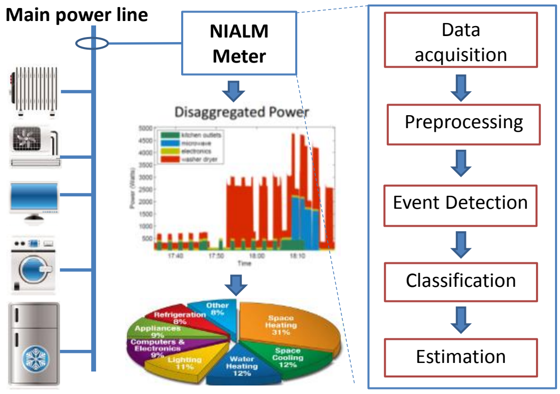


Fig. 1. The operations of NIALM meter with five basic functions

In previous research, we have developed two innovative algorithms CUSUM event detection and Genetic Algorithm for power disaggregation [6, 7]. These algorithms can detect most of events in electrical network and classify appliances almost in real-time [8]. In SAME 2014 [2], we also presented an innovative power sensor associated with these two algorithms to improve the event detection and extract electrical signature. In next section of this paper, we will present the development of NIALM base on SoC to be able to integrate our sensor to a compact and low cost NIALM sensor node.

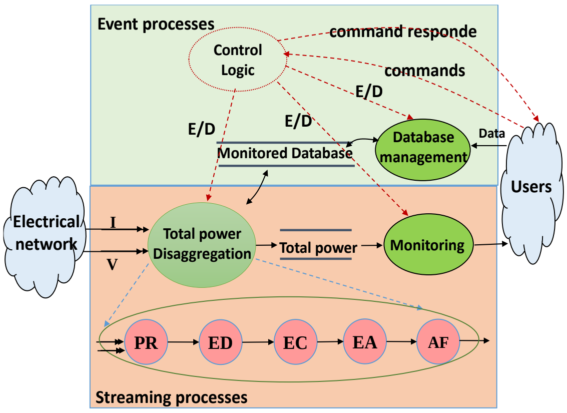


Fig. 2. Activity diagram of the system with total power disaggregation in stream and event processes for the operation mode selection

II. System Development of NIALM

A. System Specification

The aim of the modeling activity is to define the relationship between data, events, information items and the internal activities of the system. As shown in fig.2, there are two main operation modes in NIALM system the maintenance operating mode, and the power monitoring mode. System from the view of user entity works as a reactive process network (RPN) system which will respond to events generated by user's activities. Therefore, the ***control logic*** is used to enable or disable each function and display into GUI. Moreover, an important requirement for power monitoring mode is the ability in detecting an appliance in 5 seconds after its ON-OFF event.

B. Executable Specification with StateChart

The services executions in NIALM system can be either dependent (precedence, mutual exclusion) or independent and concurrent. We then model these services using the Statecharts formalism in Labview as illustrated in fig.3 with three main services: power monitoring, database management and data logger. This model one correct, can be executable so that customer can verify system activity with their requirement easier. In multi-threat processing, StateChart can model the overall activity of complex systems whose services can run independent or parallel. Moreover, Labview StateChart can automatic generate synthesizable models to rapid prototype a RPN system.

C. Architecture design using SDF model

Kahn Process Network (KPN) is often used to model the overall activity of data streaming system. However, this model does not express the rate and the quantity of token in each process so that it is difficult to analysis the timing and resources consummation to optimize these resources. Moreover, basing on data-driven scheduling, the capability in managing FIFO size at run-time may also cause the memory overflow problem in system. Synchronous Dataflow (SDF) can overcome this limit by making the static the system schedule and buffer size in communication channels for heterogeneous system in compiling time [3-5]. A SDF model requires all processes (or actors) of SDF must fire (or write to communication channels) a constant number of tokens in every firing. Therefore, SDF is the best candidate to model this NIALM system.

From the system requirement of the system, the SDF model of the disaggregation streaming function was developed as in fig.4. This model presents clearly that the system has 3 frequency domains. The measurement is invoked in a defined sampling rate F1 (5 kHz). The preprocessing is invoked periodically when N1 samples are measured from the acquisition. The Event Detection needs to check every preprocessed data from Preprocessing so that it works in the same sampling rate of Preprocessing. Finally, the disaggregation

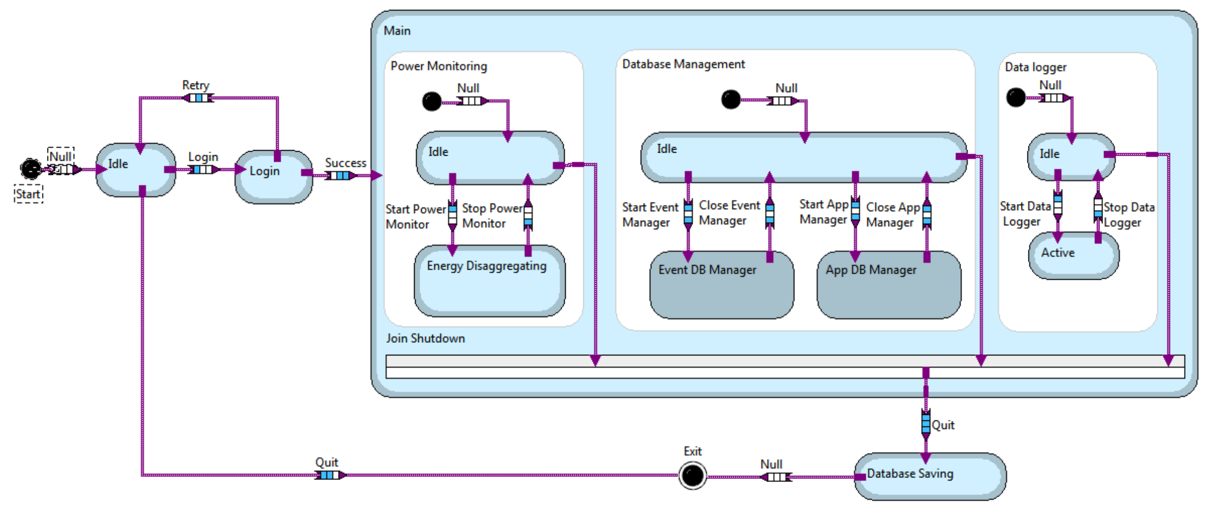


Fig. 3. The macro-state (FSM) model of NIALM system in StateChart

should be an event-based process because it is invoked when an event is detected.

Memory allocation

FIFO memory must be used in multi-rate domain system to transfer data synchronously between processes. This SDF model allows us to optimize memory size for the FIFO communication between processes as defined in table I. In this table, “b” is the preventive buffer of FIFO for the FIFO-writing can continue when the system waits to the FIFO-reading; however a FIFO-reading must always finish before a FIFO-writing.

TABLE I. MEMORY ALLOCATION FOR FIFO CHANELS IN NIALM SYSTEM

Communication Chanel	Memory requirement (Bytes)
Measurement - Preprocessing	(N1 +b)* { Sizeof (Datatype of voltage) + Sizeof (Datatype of current)}
Preprocessing – Event Detection	(1+b)*{2* Sizeof(Datatype of P/Q) + 2*k* Sizeof(Datatype of Ak/Bk)}
Event Detection - Disaggregation	(1+b)*{Sizeof(Datatype of dP) + Sizeof(Datatype of dQ) + Sizeof(Datatype of dTHDi) }

Through put analysis

Moreover, notations D_P, D_{ED}, D_{EC}, D_{GA}, D_{AF} present the worst processing tasks latency that can be set bounded values to satisfy the real-time constraints of the system. As shown in fig.4, the D_P must be less than N1*Ts where Ts is the sampling period of acquisition process. D_{ED} must be less than 20 millisecond-sampling period of Preprocessing. Finally, total of D_{EC}, D_{GA}, D_{AF} must be less than 200 milliseconds to avoid the overflow memory trouble when there are too many events detected in every 200 milliseconds. These information in SDF model can conduct the architecture exploration to minimize the processing delay in order for the system to be able to reach the timing, performance constraint.

D. Prototyping NIALM on SoC Zynq

The Zynq platform

SoC Zynq platform is selected to explore NIALM architecture because of some reasons. First, such a platform contains two technologies: the processor unit dual-core ARM Cortex A9 integrated with two media-processing engine and the programmable logic unit with rich FPGA’s resources. Second, the programmable logic of Zynq allows high-level synthesis approach in C/C++ and model-based design. Third, there are a lot of peripheral units that necessary for NIALM system such as dual 12-bit 1MSPS ADC, low-power gigabit transceivers, on chip memory, timer etc. Therefore, Zynq architecture can give several architectures to explore NIALM system: single processor, dual-processors, processor with FPGA hardware acceleration.

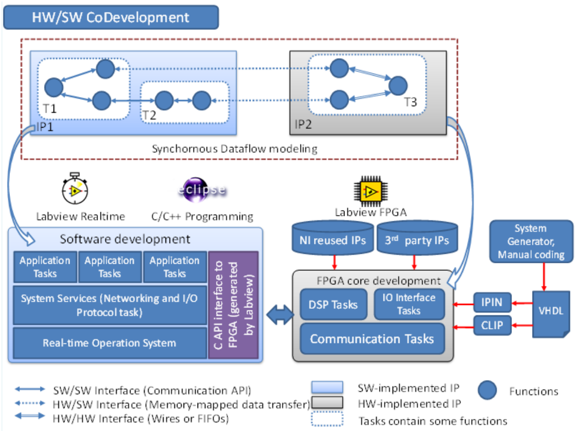
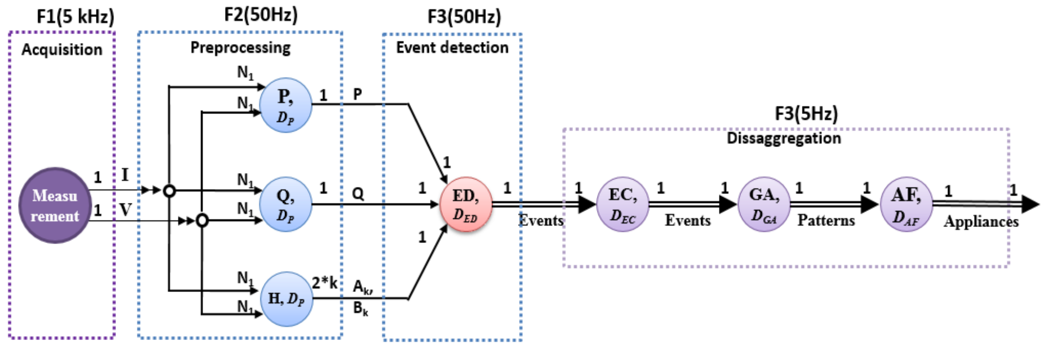

HW/SW CoDevelopment
T1
T2
T3
IP1
IP2
Synchornous Dataflow modeling
Labview Realtime
C/C++ Programming
Labview FPGA
Software development
Application Tasks
Application Tasks
Application Tasks
System Services (Networking and I/O Protocol task)
Real-time Operation System
C API interface to FPGA (generated by Labview)
NI reused IPs
3rd party IPs
System Generator, Manual coding
FPGA core development
DSP Tasks
IO Interface Tasks
Communication Tasks
IPIN
CLIP
VHDL
SW/SW Interface (Communication API)
HW/SW Interface (Memory-mapped data transfer)
HW/HW Interface (Wires or FIFOs)
SW-implemented IP
HW-implemented IP
Tasks contain some functions
Functions

Fig. 5. Hardware software co-development approach

Labview FPGA

Fig.5. presents the hardware software co-development methodology for SoC with FPGA acceleration system. Labview FPGA allows developing in parallel such a complex system from the SDF model with software, hardware and the HW_SW interfaces. This approach gives system designers the capability to quickly prototype the



P: Real power computing
Q: Reactive power computing
H: Coefficient of Harmonics computing
ED: Event Detection
EC: Event Clustering
AF: Appliance Finding
GA: Genetic Algorithm

Fig. 4. SDF model of the Total power Disaggregation function where DP, DED, DEC, DGA, DAF are times to finish processes Preprocessing, Event Detection, Event Clustering, Genetic Algorithm, and Appliance Finding

system in various hardware architecture to boost the hard real-time constraints satisfaction by the FPGA and the CPU cooperation, and to the time to market.

E. Results and conclusion

In order to select the best architecture for the system, we investigated each unit in many architecture to evaluate their performance in timing, hardware resource usage and power consumption.

	Throughput (worst case)	Resources usage
Acquisition		
FPGA (Labview based)	3.925 µs (≈255 kHz)	1106 FFs (3.1%) 672 LUTs (3.8%)
Event Detection		
1 ARM Cortex A9 667MHz (Labview RT)	91 µs (per 20ms average sample)	16.54 kBs RAM
FPGA (Labview based)	40 MHz (25 ns)	938 FFs (2.6%) 826 LUTs (4.7%)
FPGA (System Generator based)	14 MHz (71 ns)	3016 FFs (8.6%) 3924 LUTs (22.3%)
FPGA (Manual coding)	12.9 MHz (76 ns)	126 FFs (0.4%) 568 LUTs (3.2%)
Preprocesing (P,Q + Harmonic Computing)		
1 ARM Cortex A9 667MHz (Labview RT)	188µs + k*333 µs (*) (40 tokens)	16.8 kBs RAM
FPGA (Labview based)	250 ns (per token)	808 FFs (2.3%) 735 LUTs (4.2%)
Disaggregation		
Computer Intel core i5-2410M 4CPUs 2.3GHz	4 ms	847 kBs RAM
1 ARM Cortex A9 667MHz (Labview RT)	953 µs	8.03 kBs RAM

(*) k is the number of harmonic.

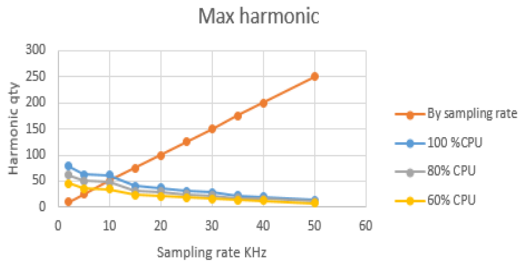


Fig. 6. Max number of harmonic analysis

Results in Table 2 show that Preprocessing in CPU can process 40 samples in 188 microseconds that means it supports about 212765 samples per second while our system acquires data in 5000 samples per second. However, Preprocessing in FPGA supports data sampling rate at 4 MHz. Unfortunately, maximum data sampling rate of Acquisition in Zynq SoC is 250 kHz (round value) that can acquire 5000 samples per a period of 50Hz electrical signal. However, the FIFO controller supports the deep only 1024 samples then maximum sampling rate is only 50 kHz. Fig. 6 shows that the best selected sample rate should not be larger than 10 KHz because CPU cannot extract the maximum harmonic. This figure also shows that the optimum number of harmonics system can extract is 50 at 10 KHz sampling rate in the one ARM architecture case. Fig. 7 presents the implementation of CUSUM event detection in this approach.

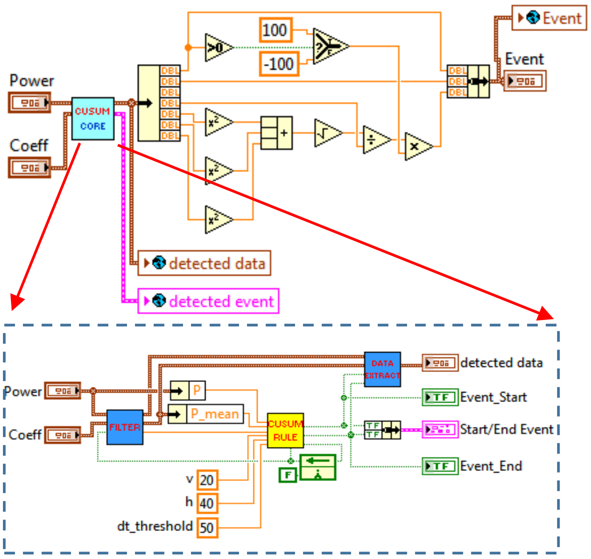


Fig.7. Modeling the event detection in LabVIEW

In coming works, we will investigate the NIALM design in complex architecture with co-operation of multi-processors and FPGA. Then power consumption of system implemented in various architecture need to be measured and compared.

REFERENCES

[1] Outlook, A. E. (2013). US Energy Information Administration Washington.

[2] Kien Nguyen Trung, Cyril Jacquemod, Eric Dekneuvel, Benjamin Nicolle, Olivier Zammit, Cuong Nguyen Van, Philippe Lorenzini, Gilles Jacquemod,”Innovative Current Sensor and Event Detection Algorithms for NIALM Application”, SAME 2014, Nice, France. Best demo and poster award.

[3] Lee, Edward A., and David G. Messerschmitt. "Synchronous data flow. “Proceedings of the IEEE 75.9 (1987): 1235-1245

[4] Zhou, Zheng, et al. "Scheduling of Parallelized Synchronous Dataflow Actors for Multicore Signal Processing." Journal of Signal Processing Systems (2014): 1-20.

[5] Bhattacharyya, Shuvra S., Praveen K. Murthy, and Edward A. Lee. "Synthesis of embedded software from synchronous dataflow specifications." Journal of VLSI signal processing systems for signal, image and video technology 21.2 (1999): 151-166

[6] Trung Kien Nguyen et al., "Using FPGA for real time power monitoring in a NIALM system," Industrial Electronics (ISIE), Taiwan, May 2013

[7] K. Nguyen Trung, O. Zammit, E. Dekneuvel, B. Nicolle, C. Nguyen Van & G. Jacquemod, “An Innovative Non-Intrusive Load Monitoring System for Commercial and Industrial Application”, IEEE International Conference on Advanced Technologies for Communications, Hanoi, 2012, pp. 23-27

[8] Trung Kien Nguyen et al., “ Event Detection and Disaggregation Algorithms for NIALM System”, NILM Workshop, Austin, USA, June 2014