IOP Conf. Series: Journal of Physics: Conf. Series **803** (2017) 012075 doi:10.1088/1742-6596/803/1/012075

The processing of information from sensors in intelligent systems

V A Kokovin and A N Sytin

State University “Dubna”, Branch “Protvino”, 9, Severny pas., Protvino, 142281, Russia

E-mail: kokovin@uni-protvino.ru

Abstract. The article describes the processing of information obtained from sensors in intelligent systems. The paper analyzes the need of advanced treatment for a paralleling operation calculator which reduces the time of response to input events. A realization of a speculative processing algorithm in the FPGA by streaming control is based on a data flow model. This solution can be used in applications related to telecommunications networks of distributed control systems.

1. Introduction

Analysis of solutions in the field of process automation (PA) of large industrial or scientific complexes, where there is a large territorial and algorithmic distribution of the individual subsystems, provides a clear choice of these process control systems – an intelligent distributed control system (IDCS). Intellectualization of the control system is not using a formalized management model, but a model based on the receipt of the information about the real outside world through sensors and the system on the basis of this information, which has the ability to adapt to external changes. In addition, an important feature of the intelligent control system is the ability to change the properties of the control devices in real time.

For example, modern cyclic particle accelerators, integrated in large accelerator structures, being one of the main tools in basic research of high-energy physics, have complex, distributed and computerized control systems. In addition, each accelerator is equipped with some means of timing required to provide a real-time mode of the control system and synchronization of technological processes in the accelerator complex. The distinguishing factor of such complexes is that the processes of the individual subsystems [1, 2] are interrelated in terms of solving the main task - to obtain a stable beam of charged particles with the specified parameters.

A survey of a large array of sensors, subsystems and technological limitations of the time spent for transformation of the information received into the events makes it necessary to use integrated networks with distributed intelligent sensors. However, this solution allows one to attain the gain in the speed only at the stage of preliminary transformations of the physical parameters of the PA into an encoded data sequence. Further, the stage of the conversion of data into the events and messages will be the "bottleneck". A good solution to this problem may be the real-time paralleling of processing of data, obtained from sensors based on Field-Programmable Gate Array (FPGA). Using the element base on the basis of FPGA for solution of management tasks allows scaling of encoded information, parallelization of data streams obtained from sensors to improve performance and a real-time control process for processing data streams in a FPGA.

2. The choice of the model for processing of the information flows outcoming from sensors

The task of raising the performance of control systems is achieved, both by increasing the computing resources and by using more effective methods of information processing.

2.1. Comparison of computational models

The use of the classical model of von Neumann computations [3], when a large amount of input data streams is ineffective, since the "narrow" place becomes a serial data processing. In this case, it is necessary to increase the clock frequency of the calculator, which is not always justified in terms of increased cost of implementing solutions. Another option - the development of architecture of the calculators on the basis of computational model ***data flow***, proposed in the works of D. Dennis, for example [4]. In this case, the implemented algorithm is "sewn up" in the equipment management system in the form of required computational and logical operators, which allows parallelizing various processes and unlimited scale.

One of the main indicators of the functioning of IDCS is the reaction of the system to the change of the state of the input event. Modern technological processes are characterized by a large number of parameters (channel) to be analyzed. However, the information, which is continuously incoming from the sensor, is not always events and requires preprocessing. It may be a situation, when an event is a function of the time intervals of incoming information rather than the information itself, or even the lack of information. Finding solutions is subjected to a predetermined logic processing of input data streams and requires significant computing resources and energy.

2.2. Selection of processing models

There are systems, such as navigation aircraft systems, which have limited capabilities of the enumerated resources. In addition, many embedded systems, process control (for example, technological systems accelerators [1]), should have a very short response time. The experience in the development of real-time control systems for power converters of physical units [1] shows that the provision of the required reaction of the control system to the events (signals) of the fast processes can only be realized by the architecture with maximum parallelized processes. In this case, it is more efficiently to apply the model of stream processing, when it becomes possible to parallelize the processing of individual channels of incoming information and to hardwire the processing algorithm. The most efficient result, taking into accounts these requirements and conditions, gives a solution based on the use of the method of streaming control.

2.3. Streaming control method

Until recently, the implementation of the effective in performance, but highly specialized in terms of function task has remained the disadvantage of the ***data flow*** model. When changing the algorithm of the task, the hardware of control systems has to be also upgraded. With the advent of FPGA technology, the situation has changed. As it was mentioned above, the narrowest bottlenecks in the processing of large amounts of data, arrived from sensors, and formation of the rapid response of control systems are a sequential execution of calculators' instructions, applied in IDCS. The solution to this problem may be the maximum parallelization of data processing and execution of the algorithm, which is the aim of the stream control method. The Stream Control Method (SCM) is based on the classical models of management that solves the problem of providing the required output parameters of the control object (process, installation, etc.) through the formation of control actions based on analysis of the information arrived from the sensors, controlling the state of the control object.

Data streams and control actions underlie the SCM, and the information from the sensors being processed based on the ***data flow*** model and the control algorithm being implemented using hardware, not software. The feature of the SCM consists in parallel processing of data streams and parallel formation of control streams.

Any control system, including based on SCM, characterized an execution model. Since the basis of the SCM is flows, the execution model, representing the SCM, is convenient to consider using DFD (Data Flow Diagramming). The diagram allows one to analyze the information flows, entering the control system input, and the control flows, formed at its output. In addition, one can use the DFD for the development of requirements specification to create control systems. Classical diagram DFD is designed for structural analysis and design of information systems. There are several commonly used notations DFD, different syntax. Diagrams describe the sources and data streams are processes and memory for storing processing results of data flows in the processes. At the same time, functionality diagram does not always correspond to the solved problems, which forces modification of DFD [5]. These diagrams are not directly used in the analysis of flow control systems with the use of the SCM for the following reasons:

- Treatment processes data streams must be strictly determined.
- Every process usually runs on a certain condition; therefore, at the process input, it is necessary to determine the logic of condition, forming the signal for initiation.

A diagram for the analysis of flows is suggested, where the elements of the *event*, the *process*, the *capsule* are used as the key elements with the assumptions described above. The word "*event*" in the system of technological object control will imply a limited set of named signals, the active state of which affects the processes. *Events* do not have the time length (in case of some action), otherwise we can talk about two *events* of the action interval: the start of the *event* and the *event* of its termination. *Events* may be external to the elements of the diagrams, or internal, if they are the results of performing the processes. We define the *process* as a set of P of the P = (S, $s0$, F)-kind, the components of which have the following meaning:

- S - set whose elements are called a P-state process,
- $s^0 \in S$ - a special state called the initial state of P,
- R - a subset of the $R \subseteq S$-type whose elements are called transitions between states.

Each *process* converts input data streams into output ones according to a specified algorithm, thus forming the control signals (exposure). Conversion of flows in the processes is carried out on the basis of the model of a finite state machine. The transition from one state into another is due to the appearance of the *event* (external or internal). *Capsule* is a structural abstraction that integrates processes implementing the input processing algorithm and control algorithm. We can say that the algorithm is encapsulated in this structure. *Processes* in the *capsules* are carried out consistently and run through the logic conditions. *Capsule* can be nested, i.e. possess a hierarchyare.

In [6], an adaptive control system of the power converter is considered, which is implemented on the basis of the SCM. The flow diagram has been developed to analyze the information and control flows presented in the form of interconnected capsules, in which a control algorithm is encapsulated. Fig.1 shows a flow diagram of the system the power converter control. The information arrived from sensors is processed by ADC and data stream enters capsules P1, P2, P3 and P4 for processing. The objective of the processes in these *capsules* is to implement the algorithm of comparing the incoming data to the set point in the case of performing the specified comparisons conditions to form event sf and control signal F to change the oscillator frequency. *Capsules* P6-P10 perform the processes of converting the information from sensor D and reference register RR for generating control signal A at input DC of the converter. Signals *clk* and *str* are external *events* to synchronize all the processes with the control system. The diagram in Fig.1 gives a visual representation of the processed streams and the generated control signals. The diagram can have a hierarchical structure and have a lot of processes. In this case, the diagram can be divided into levels and individual fragments can be visualized.

3. Speculative processing of information obtained from sensors

When solving the problems of automation of interrelated processes, a large number of cross-links appear between distributed subsystems. Telecommunication networks that have a fixed transport delay are used for delivery of the information from sensors in the form of a events and messages. In this case, the increase of the processing speed may significantly affect the IDCS performance.

3.1. Analysis of the processing time of the data obtained from sensors

Fig. 2 shows IDCS structure consisting of three sub-systems, each of which includes the following blocks: *ISN* - intelligent sensor network, *DAS* - data acquisition system, *LCS* - local control system and a power convertor. A telecommunications network, connecting subsystems, has a ring structure. Let us consider

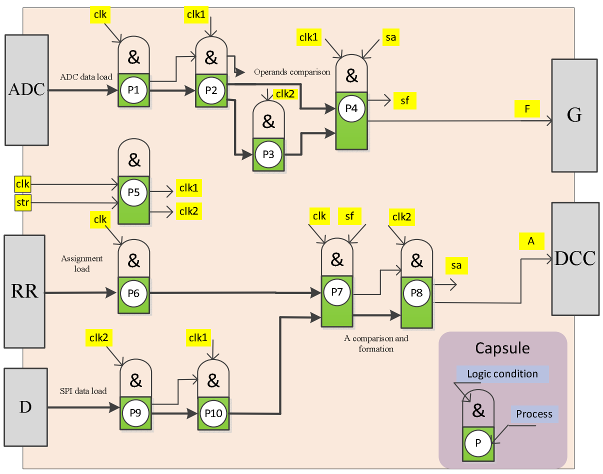

Figure 1. The diagram of information and control flows.

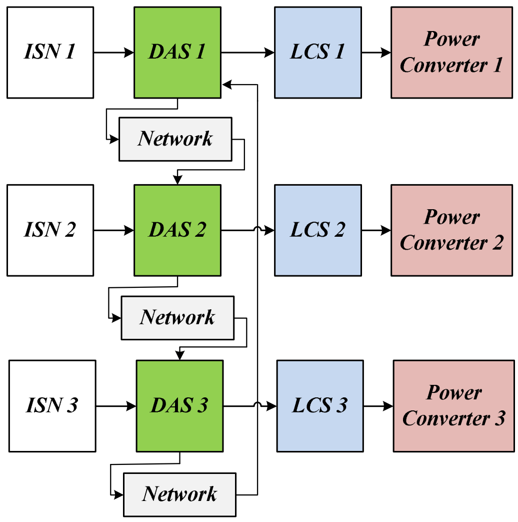

Figure 2. Directions of information dissemination in the IDCS structure.

what constitutes the delivery time of events or messages from one subsystem to another. Let the time of information dissemination $t_{propagation}$ from one subsystem to another is summarized as:

$$t_{propagation} = t_{is} + t_p + t_{tn} \quad (1)$$

when t_{is} - the time of initial information processing (for example, a temperature parameter) by intelligent sensors, t_p – the time of information processing by subsystem *DAS*, obtained from intelligent sensors, and the time of preparation for transmission via a telecommunications network, t_{tn} - the transmission time of messages via a telecommunication network. As in equation (1), time intervals t_{tn} and t_{is} are almost constant, it is possible to optimize $t_{propagation}$ only by reducing t_p.

3.2. Speculative methods of eventing

Time t_p depends on the number of sensors, realization of the calculator in the *DAS* structure, i.e. the model of processing the data obtained from the sensors, as well as sequencing of incoming messages. When using ***data flow*** in model *DAS*, when it is possible to perform scaling processing of information from the sensors, value t_p depends mainly on the speed of realization of the ordering algorithm. Articles [7, 8, 9] propose an adaptive method of eventing (adaptive speculative processing of Out-of-Order Event), arriving from the sensors, using buffering. The method is based on the assumption that processed the events that have come first, even if they violate certain order. To restore the proper order of events entering buffering. The method is based on the assumption that those events are processed that have come first, even if they violate a certain order. To restore the proper order of events entering the control system through the network, a buffer was introduced, in which events are retained for a certain time. This method significantly reduces the processing time and allows more uniform loading of the telecommunications network.

The second area associated with the use of advanced processing of the information obtained from the sensors refers to optimization of the state change of finite automata.

Article [10] presents a hardware method of optimization of transitions FSM (Finite State Machines) between states on the basis of statistical analysis of the states of automaton. This method gives a good saving time result, but cannot be realized in the SCM, as the transitions of the automaton calculator are strictly defined when synthesizing the project in the FPGA.

4. Realization of the method of speculative processing

When receiving information from a large number of sensors, the problem, concerning receiving parallel streams of events (if intelligent sensors themselves convert information into events), conversion of events into a serial stream of messages by means of the *DAS* unit and its transmission to a telecommunications network, arises [1]. The solution to this problem by the *SCM* method [6] allows the simultaneous processing of all events and performing the following steps:

- transformation of events into messages along with encoding of the address of the input port of the host system;
- encapsulation of protocol and service information into messages;
- adding the time stamp of events, entering the *DAS* unit, into the messages, if a network synchronization of the operation of individual subsystems is necessary, the messages are transmitted with a fixed period of time. Article [1] describes the network with the messages transmission period of 100ms. If you solve the problem of reducing time $t_{propagation}$, the messages are transmitted to a telecommunications network as far as *DAS* is processed.

The solution to this problem is well-established, if there are no requirements to preserving the priority of data entering the input ports when transmiting via serial communication.

Figure 3 shows an example of using the method of speculative processing of event *E1*, *E2* and *E3* in the *DAS* unit. This example corresponds to a model of processing in *DAS*, presented in [11]. Let us assume that events entered the *DAS* unit in sequence *E1* → *E2* → *E3*. The converter of events, incorporated in *DAS*, has a scanning type of formation and preparation of messages for telecommunications network *TS*. That is, when one or more events enter the *DAS* input, scanning of the inputs and the formation of messages begins. Since the processing of events *E1* - *E3* is performed in the order in which position the scanner's pointer is, the event that came later may be treated as if it came first. For example (Fig.3), event *E3* entered *DAS* last, but it was converted into message *M3* and sent to TS like first. This means that according to the speculative processing method, it is necessary to minimize the downtime *DAS* and process the event, which is pointed at by the scanner. Similarly, the rest of the events is processed. To restore the correct order of the events transmitted to the TS messages, the buffer delay of the events is used in [7].

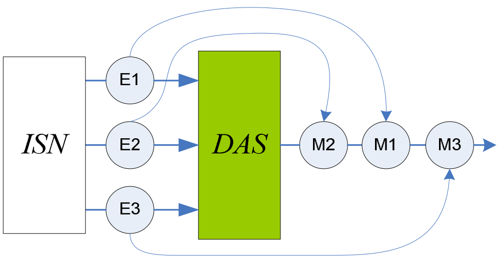

Figure 3. Speculative processing of information from sensors.

We offer to transfer the messages out of order to TS, but to restore order after their reception in the other subsystem if necessary. This solution provides the following advantages:

- in the presence of a large number of events (sensors), time data processing t_p by the *DAS* subsystem reduces significantly;
- the model of the scanning converter is implemented simply enough on FPGA;
- the method of streaming control solution, based on the **data flow** model, can be used for processing.

5. Conclusion

Trial operation of means intended for assembly and transformation of information obtained from sensors, is implemented on the basis of speculative processing in each of the subsystems of the telecommunications system [1], has shown an effective work of *DAS*. The *DAS* unit, equipped with the

64 - channel input, peak traffic > 7 Ge/s (gig events per second) and the rate of transmission of the telecommunications network equal to 100 Mb/s (UDP Protocol), is used in article [11]. Application of the speculative event processing and the buffer memory allowed a significant reduction of the time of information processing by the *DAS* subsystem.

References

[1] Ivanov S 2010 Accelerator Complex U70 of IHEP: Present Status and Recent Upgrades *Proc. of the Conf.* (RuPAC-2010, Protvino) pp. 27-31
[2] Gotsev V, Gres V, Dunaitsev A, Kim L, Komarov V and Sytin A 1993 The control and operation of the beam diagnostics for UNK *Proc. Int. Conf. ICALEPCS'93 on Accelerator and Large Experimental Physics Control Systems* (BERLIN)
[3] Burks A W, Goldstine H H and von Neumann J 1946 Preliminary Discussion of the Logical Design of an Electronic Computing Instrument *Institute for Advanced Study* (Princeton, N. J.)
[4] Dennis J 1984 Data flow ideas for supercomputers *IEEE Society, 28th International Conference* (San Francisco) pp 15-19
[5] Abedjan Z, Schulze P and Naumann F 2014 DFD: Efficient Functional Dependency Discovery *CIKM '14: Proceedings of the 23rd ACM Int. Conf. on Conf. on Information and Knowledge Management* pp 949-58
[6] Diagilev D, Kokovin V and Uvaysov S 2015 Designing a Power Converter with an Adaptive Control System for Ultrasonic Processing Units *Advances in Mechanical Engineering, Lecture Notes in Mechanical Engineering* (Springer Intern. Publ, Switzerland) pp 39-47
[7] Mutschler C and Philippsen M 2013 Reliable Speculative Processing of Out-of-Order Event Streams in Generic Publish/Subscribe Middlewares *DEBS '13 Proceedings of the 7th ACM international conference on Distributed event-based systems* (New York, USA) pp 147-58
[8] Mutschler C and Philippsen M 2014 Adaptive Speculative Processing of Out-of-Order Event Streams, *ACM Transactions on Internet Technology (TOIT)- Special Issue on Event Recognition* (New York, USA) **14** pp 1-24
[9] Hung E. and Wilton S.J.E. 2011 Speculative Debug Insertion for FPGAs *Proc. Int. Conf. FPL - 2011* pp 524-531
[10] Jackson S and Burtscher M. Self Optimizing Finite State Machines for Confidence Estimators. 2006 *Workshop on Introspective Architecture* (Austin, Texas, USA)
[11] Kalinin A Yu, Kokovin V A, Kryshkin V I and Skvortsov V V 2016 An absolute intensity beam monitor *Instruments and Experimental Techniques* **59 (4)** 536–38