VHDL Implementation of CAN Fieldbus Modeling Based on Petri nets

Hui Hu ，Ruixue Cui，Xuejie Wei
Electronics Engineering Department, North China Institute of Astronautic Engineering ,
Lang Fang 065000,China
Email: *huhui3936@yahoo.com.cn*

Abstract *–This paper introduces a process for VHDL implementation of CAN Field bus modeling based on Petri nets, which includes carrying on the model to DSPN (Deterministic and Stochastic Petri Net) of the communication behavior of CAN Field bus, combining Petri Net with VHDL language, using Petri Net to establish the model of CAN node hardware system, describing DSPN model of CAN node by VHDL language, simulating with the EDA software, obtaining the dynamic properties of the system, improving the analysis efficiency of CAN bus network performance. It is the basis of the communication protocol and circuit implementation.*

Keywords *–Petri nets,CAN bus, VHDL, DSPN model.*

I. INTRODUCTION

As the acquiring of deep submicron integrated process, asynchronous sequential circuits begin to be applied SLSI(super-large-scale integration)circuits, but there is still no breakthrough of the asynchronous circuit design. Petri net is an asynchronous concurrent system, without man-made control flow, it can express the uncertainty intuitively, and also can describe a complex system by graphic methods and analyze it by math tools. So it has been widely applied in modeling and simulation of software system.[1] VHDL is suitable for describing asynchronous concurrent system, therefore it can be associated with the models established by Petri net[4]. Use Petri nets to modeling, analyze and validate the system, then translate and edit it into VHDL code directly.

Besides the advantage of low-cost, easy development, good real time and anti-noise performance, the highest transmit speed of CAN bus is up to 1Mbps and the longest transmit distance is up to 10km, and the data traffic is reliable, that's why it has been gradually developed into field-bus which is implemented mostly in control and communications of industry departments. CAN field-bus is for industrial applications and fully distributed real-time network control system. The theory analysis and modeling of this system is not mature yet and is lack of an integrated research system. This paper carried the communication process of CAN field-bus on DSPN, and described the stochastic Petri net model by VHDL. It provides a new method to analyze the performance of field-bus communication protocol.

II. CAN BUS OPERATION MECHANISM

The operation mechanism of CAN communication protocols is that arbitrary network node can take the initiative to send a massage to other network nodes at any time and regardless of master slave .Each node has the right to use the network when the bus is idle .When the conflict is occurred ,the arbitration technology of priority of non-destructive bus is adopted .When several nodes send messages to network at the same time ,with the beginning of frame identifier， the principle of the use-by-bit arbitration is low-priority node initiative to stop sending message and high-priority node will not be affected to continue to send massage which can effectively avoid bus conflict and make the message and time to no loss. When higher-priority node is sending message ,the node will stop sending message until the network is idle again. The figure 1 is the flow chart for the communication mechanism.

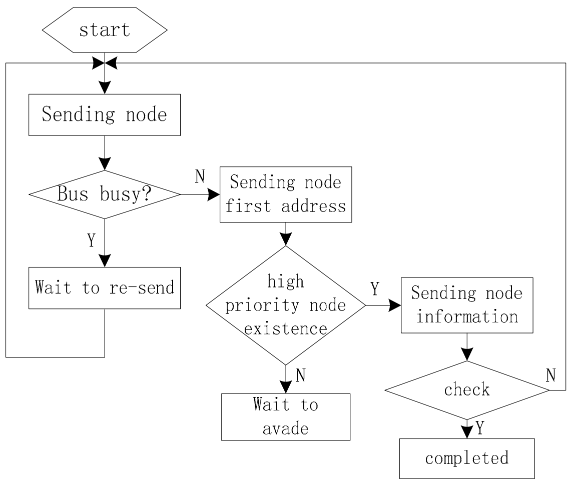

Fig.1. flow chart for the communication mechanism

III. THE DSPN MODEL OF CAN FIELD-BUS

At present, the hardware implementation research of Petri net is earlier than the development of the Petri net to VHDL compiler in China, but there are a lot more research in other countries. The Swede mapped the colored Petri net into VHDL, its main content is translating the graphic describe of Petri net into VHDL codes by translating every nodes of Petri net (store-place

978-1-4244-3864-8/09/$25.00 ©2009 IEEE

and transition) into pre-defined components of the component library. Portuguese and Spaniards developed an ECAD software. Its main part is a compiler, this software uses Cocktail compile tools to turn text description in CONPAR form of Petri net into RTL VHDL code.

Introducing time parameters to Petri nets and linking a stochastic time delay between each can be implemented and implemented transition, this kind of Petri net is called stochastic Petri net (SPN). It provides a good method to describe the system performance model. Stochastic Petri net can be approximately divided into Stochastic Petri net(SPN), Generalized Stochastic Petri net(GSPN), Stochastic Reward net (SRN) and Deterministic and Stochastic Petri net(DSPN).

The CAN CSMA/CA single node DSPN model is shown in Figure 2. In this model, store-place P indicates the system state, expressed by circle. Transition T indicates the changes of system state ,expressed by box. Generally, we use white rectangular to indicate random transition, black rectangular to indicate timing transition and line-segment to indicate immediate transition.

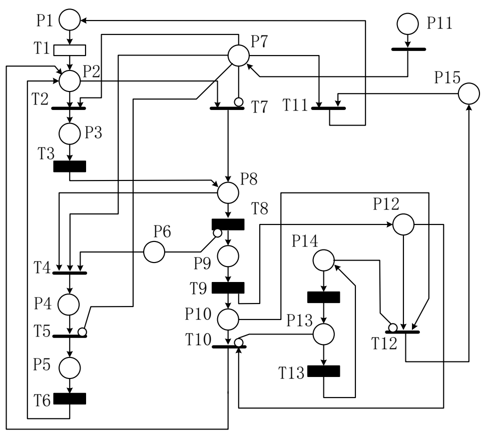

Fig. 2.DSPN communication model of CAN node

Design a can system, we must ensure the non-destroyed bit-by-bit arbitrate requirements of media access layer, namely, when bus competition is occurring, the massage with higher priority will obtain bus competition. So the physical layer must support the characteristic of CAN bus recessive bit and dominant bit. Bus is in recessive state when there is no dominant bit transmit or it is in idle mode. When one or more nodes are sending dominant bit, dominant bit will cover recessive bit which will make bus in dominant state. The meaning of state and transition of a CAN node communication DSPN model is shown in Table 1.

Every node in the net will send message to other nodes, only when the former message has transmitted successfully, then the new transmission command can be received. This model is divided into three modules: channel module, conflict processing module and normal sending and receiving module.

Table1. Meaning of states and transitions

state	meaning
P1	Readiness state
P2	Node information in queue state
P3	Waiting state
P4	Blocked state
P5	Backoff state
P6	Priority node exist state
P7	Bus channel state
P8	Send identifier state
P9	Send all information frame state
P10	Frame heck and exam state
P11	Bus channel busy-idle state test
P12	Framework check state
P13	Wrong state
P14	Right state
P15	Remove connection state
transitio n	meaning
T1	Random time transition
T2	Channel busy immediate transition
T3	Waiting timing transition
T4	Blocked immediate transition
T5	Return immediate transition
T6	Return queue timing transition
T7	Sending start immediate transition
T8	No priority conflict timing transition
T9	Waiting for inspection timing
T10	Error check immediate transition
T11	Remove channel connection and wait for new information immediate
T12	Check results correctness immediate transition

A. Channel Module

This model use store-place P7 to represent the channel state of CAN bus. When P7 takes token, means the current channel is in busy state. First, sending node is in ready state P1. After random time transition T1, node information is in queue state P2, and then sending node will check busy-idle state. If detected the channel is busy, namely P7 has got a mark, immediate transition T2 will carry out and node will enter waiting state P3 until the channel is free and after a short time delay determined by the time transition T3 parameter then transmission will start again. Otherwise immediate transition T7 will carry out and sending node will send its information frame identifier part P8.

B. Conflict Processing Module

The model use store-place P6 to represent the existence state of high priority node, when there has token in store-place means there are high priority node information sending. If detected marker contained in P6 during the period when node is in P8 state means there have high priority information transmitted in the channel. It will enter blocked state P4, experience back off state P5 then go back to P2 state again. Otherwise it will transmit all information P9 of this frame.

C. Normal Sending Module

When node is in P9 state, the node is waiting for frame check result state P10. If this frame transmission is wrong, immediate transition T10 will carry out and sending node will go back to P2 state to repeat this frame transmission. Otherwise the transmission is right, immediate transition T2 will carry out and sending node will remove channel connection state P15. After an immediate transition, sending node T11 removed the connection it will go back to P1 state, wait for new transmit requires[7].

IV. VHDL IMPLEMENTATION

After its naissance, VHDL was listed as IEEE 1076 standard. It then developed into industry standard quickly and was widely adopted by hardware designers. VHDL provides plenty of methods to generate executable hardware system description model easily. It has many abstract levels description ability and supports any mix description between abstract levels. Combined system-level modeling method with VHDL can not only use the abundant support tools of VHDL to realize system hardware design automation but can also use its simulation environment to make simulation analysis of the whole system model, thus provides rapid prototype design support of system design.

Same as Petri net, VHDL is a concurrent language. Use Petri nets to describe, analyze and verify the system design model, then turn it into VHDL description and finish the emulation and synthesis. This will make Petri net and VHDL complement each other and further analyze and verify the system performance.

A. VHDL entity description

Program entity description is to make interface description for design entity and external circuit. Choose input and output signal according to the variable which is pivotal in system protocol to ensure the VHDL description entity and its interface. In CAN bus protocol system model, channel state is designed into entity input signal and system conditions are designed into entity output signal.

B. Program structure description

Structure is used in describing logical structure and function of design entity. Based on system protocol, we can adopt multi-process structure to describe system. Multi-process structure is a parallel execution process net, multiple process execute concurrently. According to the system topological structure, each process can be mapped into system store place state, thus can describe asynchronous concurrent relationship between each store place. Processes use signals to communicate with each other, these signals are generated by process output results which are caused by new parameters. Design process program and make the generated outputs into new store place state, thus affect the transition's condition and use the output as communication signal.

C. System resource conflict

As a language which supports many description methods such as behavior description and structure description ,VHDL's concurrent statements is used in describing complication relationship and its sequential statements can describe sequence constraint mechanism. It provides a feasible method to solve conflicts inside the system. In the real Petri net model there may exist conflict, but it can't provide a solution itself[2].

VHDL's concurrent statements can not solve conflict problem, use concurrent statements to describe conflict may cause resources loss. But if use sequential statements such as the internal statements and setting different priorities can solve the existed conflict in Petri net.

D. Time delay of VHDL

During the system design procedure, it is required to build time delay model for system time delay characteristic analyzing. Bring time delay Petri net into system is crucial to system modeling and analysis. Due to its strong circuit describe and modeling ability, digital circuit can be modeled and described from system layer, algorithm layer, register transmission layer, logic layer and circuit layer. So using VHDL to describe time delay Petri net and making simulation then getting its performance characteristics will build the foundation for system design[1].

Wait and after statement of VHDL can be used in circuit time delay modeling and simulation. Wait statement is used in processes and after statement is used in realizing device transmission delay and inertia delay.

E. VHDL program implementation of CAN single node DSPN model

Part of the VHDL program is as follows:

```
ARCHITECTURE     a  OF petri1 IS
  signal ra1,rb1,r1,r2,r5,r10,ra2,rb2,rc,r8,r9:std_logic;
  begin
  process(reset)
  begin
   if reset<='0' then
```

```
    ra1<= '0';
    elsif (reset'event and reset='1' )
    then ra1<= '1',  '0'  after 10 us;
  end if;
  end process;
  rb1<=(ra1 or rc);
  process(reset,rb1)
  begin
   if reset<='0' then
    r1<= '0';
    elsif (rb1'event and rb1='1' )
    then r1<= '1'after 200 us,  '0'  after 210 us;
  end if;
  end process;
  rout1<=r1;
  process(reset,r1)
  begin
   if reset='0' then
    p1<= "00000000";
    elsif (r1'event and r1='1' )
    then p1<= p1+1;
  end if;
  end process;
  r2<=(r1 or r5) or r10;
  rout2<=r2;
  process(reset,r2)
  begin
   if reset='0' then
    p2<= "00000000";
    elsif (r2'event and r2='1' )
    then p2<= p2+1;
  end if;
  end process;
……
```

V. CONCLUSIONS

After VHDL program implementation of CAN bus DSPN modeling, using modelsim to make simulation can realize model performance analyzing easily. Simulation results are shown in figure 3.

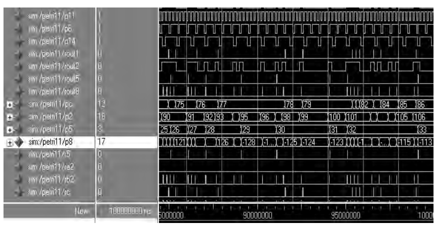

Fig. 3. Modelsim simulation results

Simulation results show that when channel busy-idle probability and high priority information sending probability are determined, as time λ (information transmission request reaching interval) increasing, bus utilization-ratio and conflict-ratio dropped. As nodes increasing, simulation time and average bus communication time delay increased. The performance analyzing results were consistent with those carried on CAN CSMA/CA single node DSPN model by WinTTPN software. It proved that VHDL can describe DSPN model well and it realized simulation by using EDA tools and obtained system dynamic performance.

ACKNOWLEDGMENT

The author wishes to thank the IEEE for providing this template and all colleagues who previously provided technical support .We are grateful to Professor Li Guohong for his constructive discussion.

REFERENCES

[1] Buhui Zhao, et al, Simulation of the Timed Petri Nets Based-on VHDL, Journal of System Simulation, Supplementary issue,pp. 99-101, Aug.2003.

[2] Fengsong Diao, et al, Translation of VHDL Description to Petri Net in High Level Synthesis , Journal of Computer-Aided Design &Computer Graphics,Vol 15, pp.1105-1111, 2003.

[3] Guohun Zhu, Congmin Wang, Optimization VHDL Design With Petri Nets Model, Journal of System Simulation, Vol 15, pp. 83-84. Aug. 2003.

[4] G. Ristori. Modeling System with Shared Resources Via Petri Nets [D]. Italy: Universitia de Pisa, 1994.

[5] R. Champagnat, P. Esteban, H. Pingaud and R. Valette. Petri netbased modeling of hybrid system [J]. Computers in Industry, Vol 36，pp.139-146. 1998.

[6] Grzegorz Andrzejewski.Timed Petri Nets for Software Applications [M]. In: Marian Adamski, Marek Wêgrayn,ed. Proceedings of the InternationalWorkshop on Discrete-Event System Design DESDes’01. Przytok: Technical University of Zielona Góra , pp.73-78. 2001.

[7] Huber, P., Jensen, K., Shapiro, R. M. Hierarchies in coloured petrinets [A]. Rozenberg, G., (Ed.), Advances in Petri Nets[C]. Berlin: Springer-Verlag, pp.313-341. 1991.

[8] Xudong He. PZ nets --- a formal method integrating Petri nets with Z [J]. Information and Software technology, Vol 43,pp. 1-18. 2000.

[9] Looney, C. G. Fuzzy petri nets for rule-based decision making [J]. IEEE Trans. Systems Man and Cybernetics, Vol 18,pp.178-183. 1988.

[10] Charles Lakos. From coloured Petri nets to object Petri nets [A]. In Proceedings of the Application and Theory of Petri Nets 1995[C]. Berlin: Springer, pp.278-297. 1995.

[11] Lakos, C. A cooperative editor for hierarchical diagrams: an objectPetri net model [A]. Proceedings of the 1st OOPMC workshop [C], pp 43-56. 1995.

[12] G. Reggio, E. Astesiano, C. Choppy, and H. Hussmann. AnalysingUML Active Classes and Associated State Machines -- ALightweight Formal Approach [A]. Proc. FASE 2000 - FundamentalApproaches to Software Engineering [C]. Berlon: Springer Verlag, pp.127-138. 2000.