
Development and Implementation of a Powertrain Electrical System Simulator with Computer-Controlled Fault Generation

Changjiang Wang and Tim Cardanha
Ford Motor Company

David D. Wentzloff
Massachusetts Institute of Technology

ISBN 0-7680-1633-9



9 780768 016338

SAE *International*[™]

2006 SAE World Congress
Detroit, Michigan
April 3-6, 2006

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE's peer review process under the supervision of the session organizer. This process requires a minimum of three (3) reviews by industry experts.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

For permission and licensing requests contact:

SAE Permissions
400 Commonwealth Drive
Warrendale, PA 15096-0001-USA
Email: permissions@sae.org
Tel: 724-772-4028
Fax: 724-776-3036



For multiple print copies contact:

SAE Customer Service
Tel: 877-606-7323 (inside USA and Canada)
Tel: 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org

ISSN 0148-7191

Copyright © 2006 SAE International

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE. The author is solely responsible for the content of the paper. A process is available by which discussions will be printed with the paper if it is published in SAE Transactions.

Persons wishing to submit papers to be considered for presentation or publication by SAE should send the manuscript or a 300 word abstract to Secretary, Engineering Meetings Board, SAE.

Printed in USA

Development and Implementation of a Powertrain Electrical System Simulator with Computer-Controlled Fault Generation

Changjiang Wang and Tim Cardanha
Ford Motor Company

David D. Wentzloff
Massachusetts Institute of Technology

Copyright © 2006 SAE International

ABSTRACT

To manage the function of a vehicle's engine, transmission, and related subsystems, almost all modern vehicles make use of one or more electronic controllers running embedded software, henceforth referred to as a Powertrain Controller System or PCS. Fully validating this PCS is a necessary step of vehicle development, and the validation process requires extensive amounts of testing. Within the automotive industry, more and more of this validation testing is being performed using Hardware-in-the-Loop (HIL) simulators to automate the extensive test sequences. A HIL simulation typically mates the physical PCS to a closed-loop real time computer simulation of a powertrain. Interfacing the physical PCS hardware to a powertrain simulation requires the HIL simulator to have extensive signal input/output (I/O) electronics and simulated actuator electrical loading.

To accomplish this needed interface with off-the-shelf I/O devices and to provide needed automated faulting of actuator loads, Ford's VPACS-HIL simulator development has led to the creation of the CAPESS electrical load system. The CAPESS is a computer-controlled, automatic, powertrain electrical system simulator. The CAPESS combines a user-defined collection of load boards in a standard rack-mount chassis. The CAPESS receives configuration and fault instructions from an external computer via a simple networking interface, but has on-board automatic protection to provide high-speed response to any failure conditions. Currently there are two forms of a load board – a standard load board and an ignition spark-coil load board. Both types of load board use solid-state circuitry, contain on-board signal processing, and allow the user to rapidly reconfigure electrical loads. Since implemented in 2000, the CAPESS has functioned very well as a key element of Ford's VPACS-HIL powertrain simulator. This paper will describe the structure, functions, and features of the CAPESS. Lessons learned in the design process and technologies employed in the design will be presented as well.

INTRODUCTION

To support the testing of complex Powertrain Controller Systems (PCS), Ford Motor Company has for some time supported the development of Hardware-in-the-Loop (HIL) test environments. In particular, the VPACS-HIL powertrain simulator is the result of extensive development efforts, with most of the early focus directed in two areas: creating powertrain models that can execute in real-time, and designing an I/O system that can properly interface to a physical PCS. The function of these primary VPACS-HIL simulation elements has been previously examined [1,2].

Another necessary element is equivalent actuator electrical loading, which presents PCS electrical sensing circuitry with appropriate electrical impedances. This avoids unintended PCS detection of electrical faults. In early implementations of the VPACS-HIL system, two standalone electrical load boxes contained the equivalent actuator electrical loads: a standard load box contained a series of fixed resistive and inductive loads mounted to boards, while a spark-coil box contained a fixed set of spark plug charging coils. Both load boxes included some modest signal processing circuitry that converted actuator driving signals to input state signals that are compatible with off-the-shelf I/O devices. As the role of the VPACS-HIL system was refined to target PCS design-validation testing, including the key area of OBD (On-Board Diagnostics) validation, it was clear that the current load boxes were not sufficient.

There were several main goals of a next-generation actuator electrical load simulator: provide the ability to generate all needed actuator electrical fault modes under computer control, create a modular and expandable load system that is easy to configure, and incorporate robust system electronics and layers of protection that will ensure long-lasting and safe operation. This next-generation electrical load simulator also had to work with existing VPACS-HIL simulators without any significant modification.

The resulting system has been named the CAPESS – Computer-controlled, Automatic, Powertrain Electrical System Simulator. Each CAPESS consists of a standard form-factor chassis containing a combination of three types of printed circuit boards: a single central processing / control board, standard load boards, and spark-coil load boards. All actuator electrical loads for a given PCS are provided by a user-specified combination of standard electrical load boards (providing resistive and/or inductive loads) and spark-coil load boards.

The remainder of this paper will detail the CAPESS' design and lessons learned during the development process, and will describe the resulting robust performance of the CAPESS during extensive use on multiple VPACS-HIL installations.

HARDWARE DEVELOPMENT

The hardware development process for the electrical system simulator had to address several important goals: functioning with all current and all foreseen future VPACS-HIL simulator designs, generating all needed actuator electrical operating modes and fault modes, flexibly providing all needed types of loads, and reliably running without human presence for many hours at a time.

Initial development brainstorming and competitive benchmarking led to selecting a hardware framework consisting of a standard computer chassis with a backplane, holding a collection of circuit boards. Each load board would contain the equivalent actuator electrical loads as well as all needed fault insertion switching and all supporting circuitry for switch control, signal conditioning, and safety protections. A single central processing board would control the operation of all load boards, based on control instructions received from the primary simulation computer.

Based on existing VPACS-HIL equipment form factors and an evaluation of backplane and circuit board space requirements, a standard 6U (1U = 1.75 inches) high, 320-mm deep, 19-inch wide rack-mount chassis was selected. Initial development intent was to develop a single load board that would support all actuator electrical loads, but unique signal conditioning requirements led to the development of two separate load boards: a standard load board is used to provide appropriate loads to most PCS output drivers, while a spark-coil load board is specially designed to load the PCS ignition drivers. Fig. 1 illustrates the resulting CAPESS board arrangement, as well as the final backplane assignment: a custom-defined VME bus was adopted for both communication between boards and for power distribution.

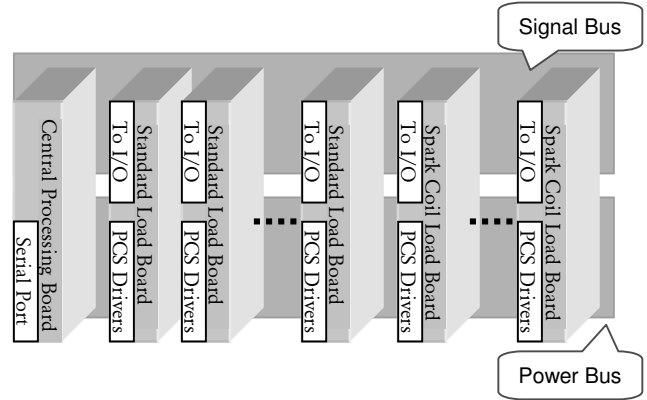


Fig. 1: Schematic of a full CAPESS installation.

For a number of reasons, including available board surface area and grouping convenience, each standard load board and spark-coil load board was laid out with eight channels. Within a given type of load board, all eight channels have identical circuit layouts. For both types of load boards, each channel has two essential elements supporting the primary load circuit: an extended H-bridge switching circuit, and signal conditioning circuitry. The common H-bridge switching implementation is detailed below, while each load board type's signal conditioning circuitry is detailed in later sections.

EXTENDED H-BRIDGE

Various PCS implementations support both high-side drivers (PCS providing a driver path to power) and low-side drivers (PCS providing a driver path to ground). In addition, as part of mandated On-Board Diagnostics (OBD), for each powertrain actuator the PCS must detect the presence of three primary electrical fault modes: short-circuit to power, short-circuit to ground, and open circuit. Selecting one of these two operating modes and three fault modes requires a switching apparatus; the CAPESS design makes use of the widely-used H-bridge arrangement.

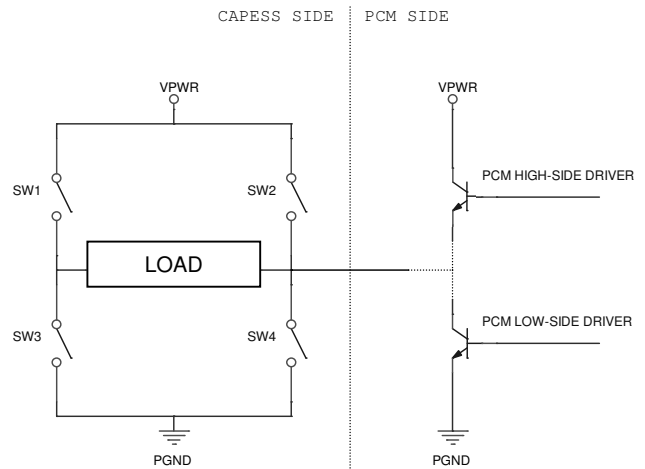


Fig. 2: H-bridge switching arrangement for a load channel.

As shown in Fig. 2, the H-bridge is composed primarily of four controllable switches SW1, SW2, SW3, and SW4. Normal operating modes or failure modes are selected by controlling the appropriate combination of ON/OFF states of these four switches, as shown in the Table 1.

Table 1: PCM Driver operating mode vs. switch states.

Driver Type	Load Work Mode	Switch Status			
		SW1	SW2	SW3	SW4
Low-Side Driver	Normal	ON	OFF	OFF	OFF
	Short to VPWR	OFF	ON	OFF	OFF
	Short to PGND	OFF	OFF	OFF	ON
	Open Circuit	OFF	OFF	OFF	OFF
High-Side Driver	Normal	OFF	OFF	ON	OFF
	Short to VPWR	OFF	ON	OFF	OFF
	Short to PGND	OFF	OFF	OFF	ON
	Open Circuit	OFF	OFF	OFF	OFF

Traditionally, mechanical relays or switches have been employed in automated electrical fault testing equipment. These electromechanical components have many drawbacks, such as short lifetime, low reliability, and slow response. To avoid these problems, the H-bridge switching circuitry for the CAPESS makes use of MOSFET and IGBT solid-state switches. To further enhance the robustness of the H-bridge switching circuitry, a gate control circuit commonly used with other applications of an H-bridge has been included, thus prohibiting short circuits resulting from simultaneously closing SW1 and SW3 or SW2 and SW4.

CAPESS BUS

As shown in the Fig. 3, a standard 6U VME chassis was selected to accommodate the CAPESS boards. The upper backplane provides a 16-bit address bus, an 8-bit data bus, a control bus, and circuit-power busses. When a control action is initiated, the address bus specifies an individual load channel, the data bus indicates what configuration action to take on the specified load channel, and the control bus synchronizes the action by coordinating various low-level "ready" states.

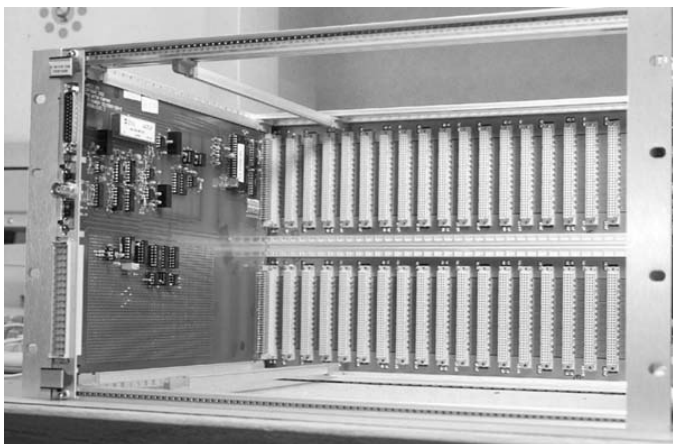


Fig. 3: CAPESS central processing board and bus backplanes.

The lower backplane provides power to the simulation loads. Two independent power busses are defined; this provides additional flexibility to support PCS applications where two separate power supplies are used to drive different actuators. This dual-power bus feature was carried through to the design of the load boards, where configuration options provide a way to select either power supply for each individual load channel.

CENTRAL PROCESSING BOARD DEVELOPMENT

In designing the central processing board, the primary goal was to create a single-point CAPESS controller with a widely used, easy-to-implement, and bi-directional communication method. Based primarily on existing VPACS-HIL communication hardware, a standard RS232 serial port connection was selected.

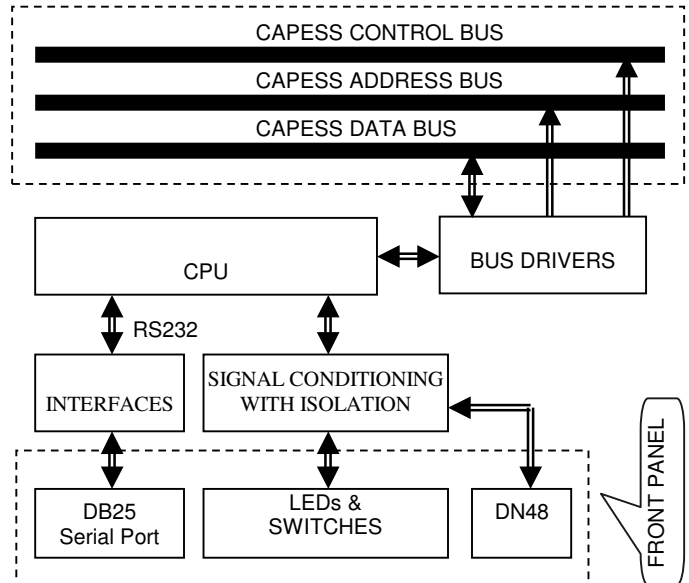


Fig. 4: Functional block diagram of central processing board.

The remainder of the central processing board's design involved designing circuitry and embedded software to translate the external communications and provide the needed control and status checking functions. Fig. 4 depicts the basic function blocks of the central processing board. Starting with the board front panel, a DB25 connector is used for the RS232 serial port link between the central processing board and the simulation computer. LEDs on the front panel are used to indicate the power and system status. The front-panel switches are used to reset the system from a failure mode condition or an unknown state.

Internally, the central processing board's CPU may access any channel of the load boards via the data, address, and control busses. This access allows the CPU, as the CAPESS master controller, to control the operating mode of each channel and to request the current status of each channel. The CPU may visit any particular channel of load or coil boards via the address bus. The upper 8 bits of the 16-bit address bus are used to define the target board, and the lower 8-bits to define

the target channel. This addressing approach can allow the CAPESS system to accommodate a total of 256 load and coil boards, with each board able to address 256 channels; overall, these addressing limits provide a significant buffer to any foreseen system requirements.

In addition to the primary function of controlling all load channels, the central processing board was recognized during development as a useful resource for a number of needed signal handling functions. Placing these functions external to the simulation computer supports the general philosophy of maintaining an off-the-shelf simulation computer system and off-the-shelf I/O devices. The functions include control of high current 'battery' power relays and conditioning of a key simulation-timing signal. Several on-board circuits, some interfacing with the central processing board's CPU, perform these miscellaneous functions.

STANDARD LOAD BOARD DEVELOPMENT

When designing the standard load board, there were many lessons to be taken from the existing load box. The original load box could not be reconfigured, which is a necessary and constant part of the testing process with frequent changes in the target PCS. The load box did not support any computer-controlled fault generation, critical for automation of OBD design validation testing. The load box did not provide any functional feedback, greatly complicating any PCS debugging efforts. And the load box contained a fair amount of hand-wiring with no safety monitoring or auto-shutoff features, compromising robustness.

The resulting CAPESS standard load board, shown in Fig. 5, is designed to load all PCS drivers except for spark ignition drivers. There are eight identical channel layouts on the standard load board. Each channel's electrical load, composed of a resistor and/or an inductor, can be mounted to the board using two pairs of mounting screws; one pair for the resistor and another for the inductor. Each channel's resistor and inductor are connected in series by a circuit trace on the board. There are two connectors on the front panel of the load board. The lower connector connects the PCS drivers to specified load channels. The upper connector routes actuator state signals, namely the analog current level and the digital ON/OFF state for each PCS drivers loaded by the board, to analog and digital input I/O devices.

To handle low-level on-board communication and control functions, each standard load board uses two CPLD programmable logic devices. For convenient board addressing, an 8-bit dipswitch is used to define each standard load board's address. When the central processing board sends an instruction out on the CAPESS signal bus, each load board's CPLD compares the desired board address to the dipswitch setting. If there is a match, the matching board's CPLD performs the specified instruction on the specified channel.

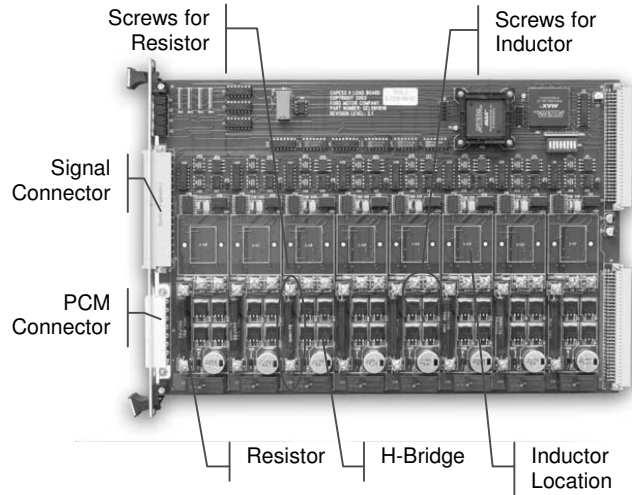


Fig. 5: Standard load board.

An instruction from the central processing board would command a load board's CPLD to either return the present state of a channel, or change the operating mode of a channel. When defining a channel's operating mode, the CPLD can specify the load power source and configure the power routing H-bridge switches. These low-level CPLD control signals and the affected switches are detailed in the one-channel schematic shown in Fig. 6. As shown in the figure, four MOSFET transistors are used to form the H-bridge; these transistors are all rated for 40A-100V, ensuring safe operation even during failure mode simulations. A dedicated H-bridge gate driver, part HIP4081A, has four control inputs (STG, STV, NHS, and NLS) that map to the four H-bridge switches. Also shown in Fig. 6, a channel's load power source is selected using CPLD signal PSL.

In addition to computer-controlled switching, the other key aspect of each standard load channel's operation is signal processing: translating higher-current actuator driving signals into actuator state signals suitable for I/O input. With the standard load board, each load channel provides the load's analog current flow and digital ON/OFF state. As shown in Fig. 6, I_C represents a single channel's sensed load current as determined by the current sensor, part HCPL788. This sensor delivers an amplified voltage signal proportional to the voltage drop across the current sensing resistor, typically 0.025Ω to provide a 10A range in sensed current. The signal I_C is routed to a front-panel connector of the load board, and is ultimately read into a channel of an off-the-shelf analog input board for use with the VPACS-HIL powertrain simulation. In addition to sensing current level, part HCPL788 is also able to detect an overcurrent condition when the sensed current exceeds a preset limit, and set a fault flag OC. The signal OC is routed back to the CPLD, which opens all H-bridge switches on the affected channel if an overcurrent condition is indicated. To determine the digital ON/OFF state, each load channel incorporates a high-speed optocoupler, part HCPL3700, that detects the voltage drop across the R-L load. The digital output of the HCPL3700, signal V_C ,

indicates the ON/OFF status of the PCS driver. This signal V_C is routed to a front-panel connector of the load board, and is ultimately read into a channel of an off-the-shelf digital input board for use with the VPACS-HIL powertrain simulation.

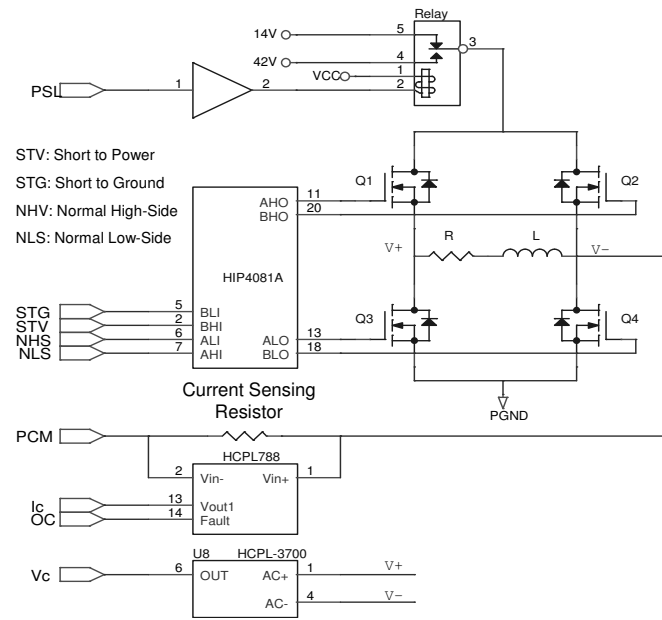


Fig. 6: A simplified schematic of one standard load channel.

A special "double-ended" PCS actuator involves a closed-loop power circuit where the PCS provides both the high-side and the low-side drive transistors. The standard load board supports this function by providing an additional switch point that connects two load channels, as shown in Fig. 7. In this mode, all switches of the two involved H-bridges should be turned off for normal operation.

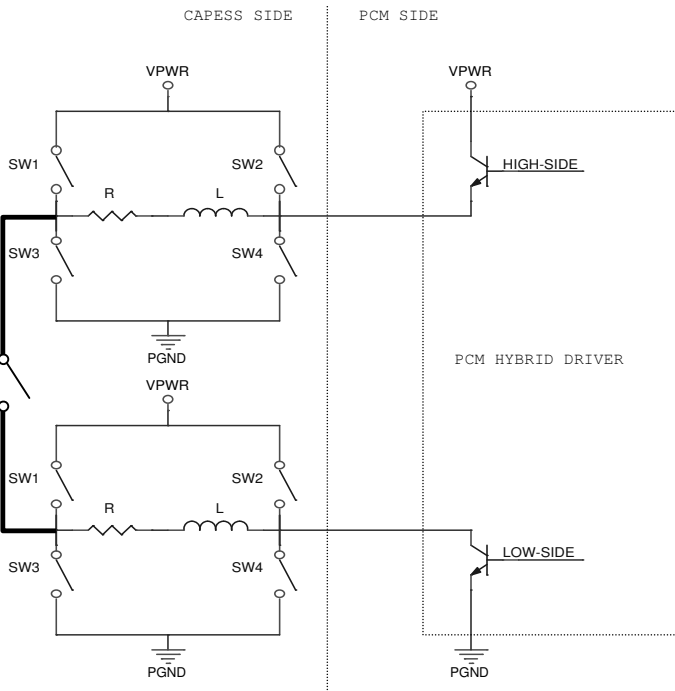


Fig. 7: PCS double-ended actuator circuit schematic.

SPARK-COIL LOAD BOARD DEVELOPMENT

As mentioned previously, loading and detecting the state of the PCS ignition drivers had always been handled by a standalone spark-coil box. Investigations of the PCS ignition circuitry, and the associated voltages generated during ignition, made it clear that a specialized load board would be needed to interface with the PCS ignition drivers.

The resulting CAPESS spark-coil load board is specially designed to load PCS ignition drivers. There are eight identical channels that are populated with simulation coils, as shown in Fig. 8. To ensure PCS ignition drivers work properly with the spark-coil load board, the simulation coils are designed based on real spark-plug charging coils: the primary side of a simulation coil has the same electrical characteristics as a real spark-plug charging coil. Therefore, PCS ignition driver fault detection circuitry won't report any electrical malfunction, and the spark-ignition driving process will occur identically to a real vehicle.

There are three connectors on the front panel of the coil board. The lower one interfaces with the PCS ignition drivers. The upper two bring out sensed current and voltage information for all channels. The interface of the board with the central processing board is managed by a CPLD that decodes commands and exchanges data with the central processing board via the address, data, and control buses. Like the standard load board, the spark-coil load board also uses an 8-bit dipswitch to determine its board address.

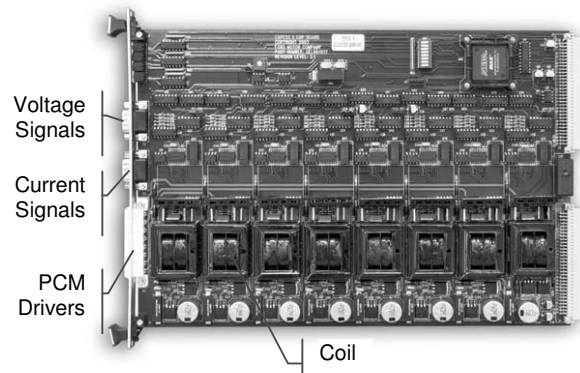


Fig. 8: Spark-coil load board.

Fig. 9 shows a schematic diagram of a single spark-coil channel, which uses an H-bridge topology that is functionally identical to that used with the standard load board. However, high voltage spikes (higher than 450V) occur in the ignition drivers during the ignition events. Only one leg of the H-bridge is exposed to these high-voltage spikes: switch Q4 in the Figure. Therefore, a high-voltage ignition IGBT is employed for the switch Q4.

In addition to handling high voltages experienced during the spark ignition process, signal processing of the more complex ignition voltage patterns required a special

solution. A specialized magneto-resistive current sensor, F.W. Bell part NT-5, measures the coil current flowing through its primary side. Its secondary side outputs an isolated voltage signal, I_D , proportional to the primary current. The level of this signal, I_D , is compared to a minimal current threshold to determine when the spark-coil is charging and when the spark-coil “fires.” The resulting digital state output is routed to a front-panel connector of the load board, and is ultimately read into a channel of an off-the-shelf digital input board for use with the VPACS-HIL powertrain simulation. In addition to sensing based on current, voltage state is detected using a high-speed optocoupler, part HCPL2631, with its input tied to the spark coil. Its digital output, V_D , reflects the voltage changes on the ignition driver and is directed to a front-panel LED.

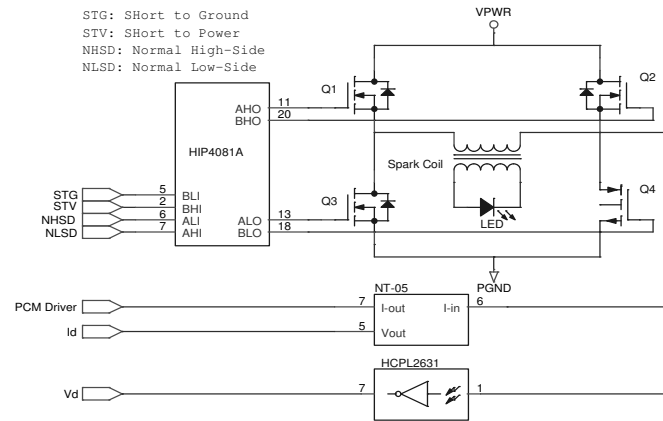


Fig. 9: A simplified schematic of one spark-coil load channel.

As noted previously, the spark-coil load board’s fault generation switching function is identical to the standard load board.

To illustrate normal operation of a spark-coil channel, Fig. 10 shows three significant signals during a single firing event. The top trace represents the voltage on a PCS spark driver, the bottom trace represents the current flowing through the coil, and the middle trace is the firing-detection signal determined from current.

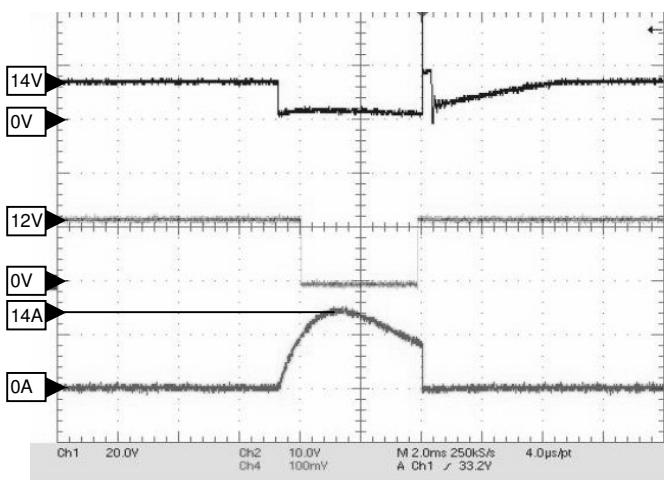


Fig. 10: Spark coil voltage (top), current (bot.), and detection.

SOFTWARE DEVELOPMENT

With the modular approach taken to the development of the CAPESS, embedded software development was needed for both the central processing board and the load boards. The majority of this embedded software development focused on the various forms of necessary communication. The central processing board communicates in two manners: with the simulation computer via a RS232 serial port, and with the load boards through the CAPESS backplane busses. Both modes of communication are handled primarily through an on-board microcontroller; the coding that manages communication and associated decision making is the major part of the software embedded in the flash memory of the microcontroller. Both types of load boards employ CPLDs to communicate with the central processing board via the CAPESS backplane busses; the CPLD programming essentially amounts to extensive switch and logic gate configuration; all of the more complex decision making is handled by the central processing board’s microcontroller.

SERIAL COMMUNICATION

The CAPESS communicates with the simulation computer over an RS232 serial port using 8-bit ASCII protocol and transmitting at 9600 bps. The CAPESS will acknowledge receipt of a command or data by transmitting an ACK (acknowledged) or NAK (not acknowledged). Fig. 11 shows the serial communication flow chart. All bytes transmitted and received over the serial port are standard ASCII characters. Addresses and Data are converted to a string of ASCII bytes and transmitted MSB first. Table 2 defines the ASCII protocol commands.

LOAD BOARD ADDRESS COMMUNICATION

When receiving an instruction from the central processing board, both types of load board use the same address decoding approach. Each channel of each load board has a unique 16-bit address, where a board ID dipswitch defines the eight most significant bits. The CPLD programming flashed to each load board configures the CPLD gates to compare these significant bits of the address bus (signals AB_x) to the bits set with the dipswitch (signals BS_x) using an exclusive-or (XOR) operation; this gate arrangement is shown in Fig. 12. The output of these operations are fed into a multiple-input “AND” operation, indicated as item 9& in Fig. 12.

The ultimate outcome of these comparisons is simply that if the specified bus address matches the dipswitch configuration, the ENABLE signal is set high (logic ‘1’), which enables the CPLD to further process the channel and action specified on the bus.

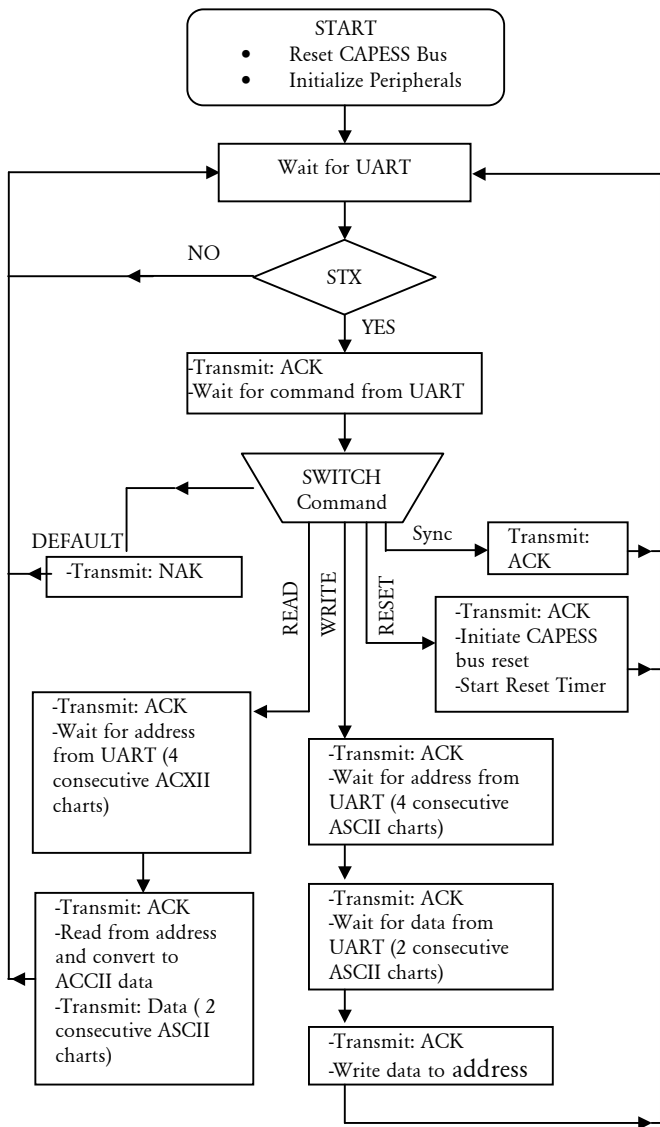


Fig. 11: Serial communication flow chart.

ASCII Commands	HEX Value	Description
STX	0x02	Start of Text
ACK	0x06	Acknowledge
NAK	0x15	No Acknowledge

Table 2: ASCII command protocol.

Note that since each load board has only eight channels, only three bits (denoted A0, A1, and A2) are needed to specify the target channel; the remaining five bits (denoted A3 to A7) of the channel-address bus should always be set low (logic '0'); a check of these extra address channels is incorporated in defining the ENABLE signal, shown at the bottom of Figure 12. The three active channel bits are combined with the ENABLE signal to specify a final target channel – this is performed with additional CPLD logic gates, not shown, following a logic conversion shown in Table 3, see appendix. In the CPLD logic, specifying a final target channel results in setting the given channel's enable (signal ENA in Fig. 13) to high (logic '1'), thus allowing control of the channel's H-bridge.

LOAD BOARD H-BRIDGE STATE CONTROL

With a channel specified, the next portion of the communication / control logic is to specify the state of the channel's H-bridge. As discussed earlier, each load channel's working mode is defined by the state of the four solid-state switches comprising the channel's H-bridge (elements Q1, Q2, Q3, and Q4, as shown in Fig. 6). Controlling the state of an H-bridge is accomplished with a combination of programmed CPLD logic and external D-type flip-flops, which latch (hold) commands from the CAPESS data bus until cleared using the /RESET control instruction. As shown in Fig. 13, four bits of the data bus (DB3, DB2, DB1, and DB0) will be latched when the channel is simultaneously addressed and enabled, i.e. both ENA and /WR are high (logic '1').

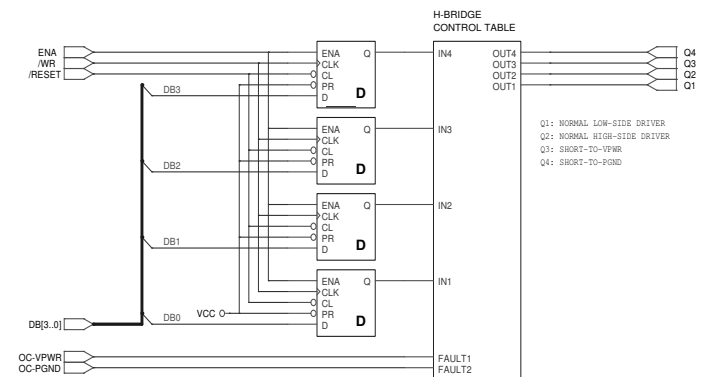


Fig. 13: H-bridge control logic generation.

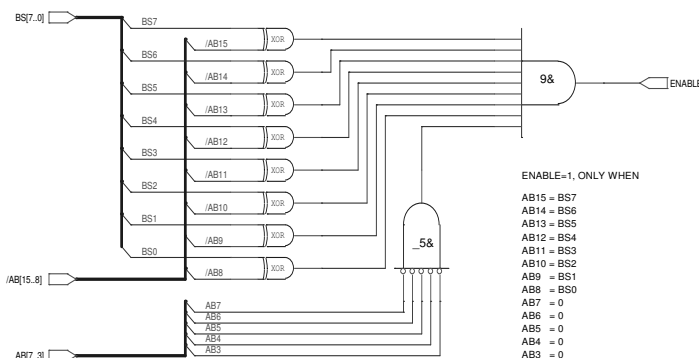


Fig. 12: Address decoding logic.

The latched states of the CAPESS data bus (IN4, IN3, IN2, and IN1) are then input to programmed CPLD logic that determines the state of the four H-bridge control signals. This CPLD logic considers these latched states as well as two fault states (FAULT1 [F1] and FAULT2 [F2]) that indicate detection of an over-current event during a prior attempt at shorting the target load channel either to power (STV) or to ground (STG). If either of these two fault indicators are set, the H-Bridge quits from the failure-mode simulation and returns to a normal operating mode. If no over-current faults are indicated, the CPLD logic defines the control signals to set the H-bridge switches appropriately; see Table 4 in the appendix for the associated H-bridge control table.

Further detailing over-current fault detection is the final topic within communication and programming logic. The two H-bridge operating modes that can result in an over-current fault are short to power (STV) and short to ground (STG). These operating modes engage H-bridge switches that intentionally bypass the equivalent electrical load in order to exercise the PCS' fault detection capabilities. However, if some element of the PCS fault detection circuitry does not function correctly, the result could be an extremely high current flow that quickly damages both the CAPESS and the PCS hardware. To prevent this, a combination of CAPESS load board circuitry and CPLD programming is used to remove the H-bridge fault mode (STV or STG) if the PCS does not respond quickly enough.

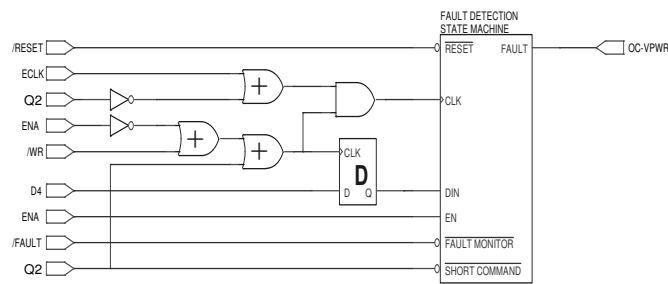


Fig. 14: Over-current fault detection logic.

To illustrate the CAPESS load board over-current handling process, consider the case of initiating a STV fault through the H-bridge, see Fig. 14. Setting a STV fault is accomplished by setting H-bridge control signal Q2 to high (logic '1'). When Q2 is asserted, i.e. the H-bridge is commanded to work in the short to power mode, the fault detection state machine is clocked (start-time is defined) by the ECLK signal. The ECLK is a clock signal whose frequency is programmed by the CPLD, typically to 10kHz. After the state machine is clocked, it will then monitor the over-current signal from the channel's current sensing chip, /FAULT, at the rising edge of each ECLK timing pulse. If the fault detection state machine receives the /FAULT signal for a duration greater than a preset time window (which gives the PCS time to react to the over-current condition, typically 200µs), the fault detection state machine sets the over-current fault (in this case, signal OC-VPWR). With this fault signal set, the H-bridge will be forced to quit the STV condition and return to normal mode, protecting the circuit from over-current damage.

CAPESS APPLICATION TO AUTOMATING OBD VALIDATION TESTING WITH VPACS-HIL

The ultimate goal of the CAPESS development project is to incorporate the CAPESS electrical simulation system in the VPACS-HIL powertrain simulator, as shown in Fig. 15, and automate testing of PCS responses to actuator electrical faults. The traditional approach to testing PCS actuator electrical faults is manual and very labor-intensive; a technician typically follows a process manual

and works in a real vehicle, switching individual switches on a signal breakout box. The time to complete this process, while varying, is usually on the order of several days to a week.



Fig. 15: A VPACS-HIL system with CAPESS highlighted.

In contrast, when working with the VPACS-HIL system, the majority of the work can be performed before PCS testing hardware is available. In the weeks before a first pass of testing is anticipated, four key preparation steps are completed: VPACS-HIL vehicle characterization data files are defined, an application wiring harness is designed and built, the CAPESS load configuration is specified, and a test sequence dataset is assembled. When a PCS is available to test and a VPACS-HIL simulator is scheduled, the simulator setup steps are completed: the CAPESS load boards are configured, and the application wiring harness and PCS are installed. For a typical vehicle application, these setup steps can be completed in a couple of hours. Initial system functional testing is then completed to confirm that the VPACS-HIL simulator system is properly configured and correctly interfaced to the PCS. Barring complications, the prepared validation test sequence can now be completed, entirely under computer control.

By introducing the CAPESS and piloting this automated testing methodology, a test sequence validating PCS response to actuator electrical faults has been completed in as little as eight hours. As part of a larger OBD

automated testing procedure that also includes sensor electrical faults, VPACS-HIL OBD Design Validation testing often runs unattended overnight. This can yield a test completion rate as much as ten times higher than is possible with technicians performing manual testing during a 40-hour work week.

CONCLUSION

Within Ford Motor Company, the VPACS-HIL powertrain simulator has been extended through the development of the CAPESS actuator electrical system simulator. The CAPESS design advances two unique concepts. First, electrical loading systems should be modular and standalone units to the greatest extent possible. This allows the remainder of HIL powertrain simulators to be designed more flexibly and with less expense. Second, electrical loading systems should be easy for the end-user to reconfigure; this allows a HIL simulator to more easily switch between different testing applications.

The resulting CAPESS electrical system simulator meets all of the design goals. It is a modular system of a central processing board and two types of load board, housed in a standard chassis. The CAPESS interfaces with an external simulation computer through simple RS232 serial port communications. Using this interface, the simulation computer can change the operating mode of any load channel or request the operating status of any load channel. If an electrical fault is initiated and the PCS doesn't handle the fault condition as intended, on-board CAPESS protection circuitry handles any over-current conditions before damage can occur. Also, on-board signal conditioning circuitry outputs the state of each actuator based on both current and voltage; these actuator states are then input through off-the-shelf I/O boards for use in the powertrain simulation.

The CAPESS system is now a key component of all VPACS-HIL systems; most of the CAPESS units have been in operation for several years or more, completing many thousands of tests. Extensive use of the CAPESS during automated actuator electrical fault testing has demonstrated both the robustness of the CAPESS architecture and the benefits of test automation. Running automated OBD test procedures, the number of tests completed per week has been as high as ten times greater than with manual testing.

ACKNOWLEDGMENTS

The authors would like to acknowledge the efforts of all the other members of the current VPACS-HIL development team, including Isis Messih, Guido Barta, Steve Cibulas, Bill Courtney, Chuck Southwell, Charlie Johnston, Kim Murphy, Ron Boudia, and Bryan Nader, as well as past team members and other contributors to the VPACS project.

REFERENCES

1. Cardanha, Tim and Jackson, Ken (2003, August). Virtual Powertrain, Real Results. *Automotive Engineering International*.
2. Cardanha, Tim (2005, April). Validating Powertrain Controller Systems with the VPACS-HIL Powertrain Simulator. *Proceedings of the SAE 2005 World Congress & Exhibition*.

CONTACT

Changjiang Wang received his Ph.D and MSEE from China University of Mining and Technology and a BSEE from Zhejiang Univeristy in China. He has worked for Ford Motor Company for five years, focusing on electronics design for the VPACS-HIL system. Email: cwang15@ford.com

Tim Cardanha received a BSME from Cornell University and a MSME from Purdue University. He has worked for Ford Motor Company for ten years, focusing on VPACS-HIL system design and application for the last seven years. Email: tcardanh@ford.com

David D. Wentzloff received a B.S.E.E. from the University of Michigan, Ann Arbor in 1999, and an S.M. degree from the Massachusetts Institute of Technology in 2002 where he is currently pursuing a Ph.D. degree in RF communication. During 1999-2000, he held internships at Ford Motor Company where he primarily focused on the development of a prototype CAPESS system. Email: ddw@mit.edu

APPENDIX

Table 3: True Value Table for Channel Selection.

INPUTS				OUTPUTS CHANNEL							
ENABLE	A2	A1	A0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0
0	X	X	X	0	0	0	0	0	0	0	0

Table 4: H-Bridge control table.

INPUTS						OUTPUTS			
F2	F1	IN4	IN3	IN2	IN1	OUT4	OUT3	OUT2	OUT1
0	0	X	0	0	0	0	0	0	1
0	0	X	1	0	0	0	1	0	0
0	0	0	X	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	1
0	0	1	1	0	1	0	1	1	0
0	0	0	X	1	0	1	0	0	0
0	0	1	0	1	0	1	0	0	1
0	0	1	1	1	0	1	1	0	0
0	0	X	X	1	1	0	0	0	0
X	1	X	0	X	X	0	0	0	1
X	1	X	1	X	X	0	1	0	0
1	X	X	0	X	X	0	0	0	1
1	X	X	1	X	X	0	1	0	0