

# Vom Chip- zum System-Design mittels SUCCESS Co-Verifikation

von Dr. Lars Voßkämper und Didier Maurer

## Inhalt

♣ Lösungen für Mechatronik	p1
♣ Anwendungsbeispiel Mikromechanik	p2
♣ Anwendungsbeispiel Makromechanik	p3
♣ Schlussfolgerung	p5
♣ Unsere Angebote – Ihre Vorteile	p6

In diesem Dokument stellen wir Ihnen Lösungen für die Modellierung von Systemen vor, die sowohl elektronische Bauteile, nichtelektronische Komponenten und Software enthalten. Wir zeigen die Vorteile der Nutzung der SUCCESS™ Co-Verifikationsumgebung:

- ♣ Bestimmung der geeignetsten Mikrocontrollervariante für eine Anwendung
- ♣ Entwicklung und Prüfung der Anwendungssoftware
- ♣ Simulation des Gesamtsystems, einschließlich elektronischer und mechanischer Komponenten und deren Wechselwirkungen, dank der Nutzung von elementaren elektromechanischen Modellen, die von Dolphin Integration bereitgestellt werden.

## Lösungen für die Mechatronik

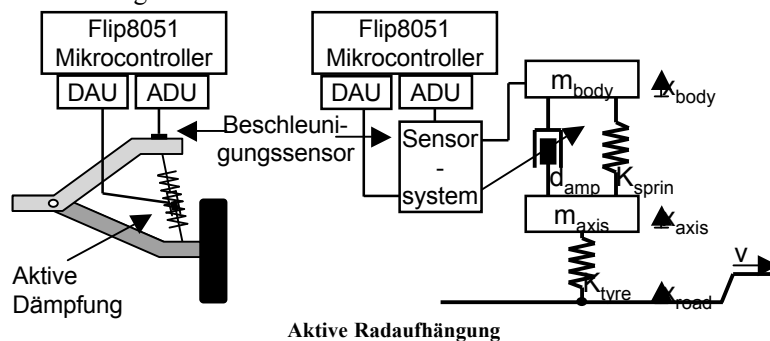
### Lösungen für Mechatronik

Das Feld der Mechatronik umfasst sämtliche Anwendungen, bei denen elektrische, elektro-mechanische und rein mechanische Elemente genutzt werden, um ein System mit entsprechend geforderten Funktionalitäten aufzubauen. Für eine schnelle und preislich vertretbare Entwicklung ist es von absoluter Notwendigkeit, neben der Konstruktion von Prototypen, sämtliche involvierte Komponenten gemeinsam, in einer möglichst effizienten Art und Weise zu simulieren. Dies ist ein scharfer Schnitt im Vergleich zu der PCB Designpraxis!

Daher versorgt Dolphin Integration Designer mit Modellen von elementaren mechanischen Elementen zuzüglich zu den elektronischen Bauteilen in der Bibliothek DIADEM(1). Die eingebundenen mechanischen Elemente bilden die Basis für die Modellierung der gesamten Mechatronik-Anwendungen. Die Bibliothek EMBLEM(1) enthält generische Modelle von Komponenten unterschiedlicher Domänen der Mechanik, Elektromechanik und Leistungselektronik.

(1) Von Dolphin Integration geliefert

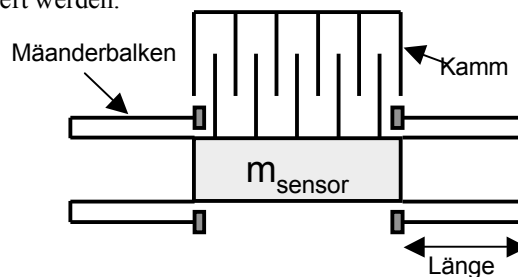
Im Folgenden ist ein semiaktives Radaufhängungssystem dargestellt, um die Karosserieschwingung während des Betriebes mittels Informationen des Sensorsystems zu kontrollieren. Das gesamte Aufhängungssystem enthält ein MEMS (Struktur eines kapazitiven Beschleunigungssensors), Makromechanik (Kraftfahrzeug) und eine 8051-basierte Controllerschaltung als ein Beispiel für die Modellierung und Simulation von Mechatronikanwendungen.



## Mikromechanischer Beschleunigungssensor

### • Aufbau und Funktion

Der mikromechanische Beschleunigungssensor ist aus einer seismischen Masse und vier identischen mikromechanischen Mäanderbalken, die als Federn dienen, aufgebaut. Die Reibung der Masse mit dem umgebenden Medium (z.B. Luft) trägt einen Beitrag zur Dämpfung in dem System bei. Ein Kamm, bestehend aus zwei Reihen versetzt angeordneter, elektrisch leitender Platten, misst die Verschiebung der Masse zwischen den Platten. Die Kapazitätsänderungen können mit einer geeigneten analogen Schaltung analysiert werden.



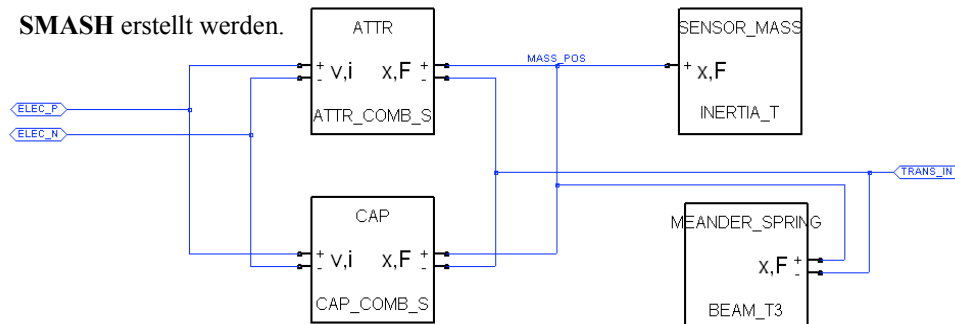
Mikromechanischer Beschleunigungssensor

### • Modellierung

Die Sensormodellierung wird durch die Nutzung der Modelle aus der Bibliothek **EMBLEM-Mecha** von Dolphin Integration vereinfacht. Diese Bibliothek bietet Modelle der wichtigsten Effekte, die in und zwischen mechanischen Modellen auftreten, z.B. Biegen von Balken, Massenträgheit sowie Anziehungs- und Kapazitätseffekte zwischen zwei leitenden Platten. Sämtliche Modelle können durch den Anwender parametrisiert werden. Mit Hilfe dieser "Basiseffekt"-Modelle ist es möglich, Modelle hoher Komplexität und jeder Genauigkeit herzustellen.

Die Modelle sind in dem IEEE Standard 1076.1 (VHDL-AMS) implementiert, welche die Möglichkeit bietet, in einem Schaltungssimulator komplette Mechatroniksysteme zu simulieren.

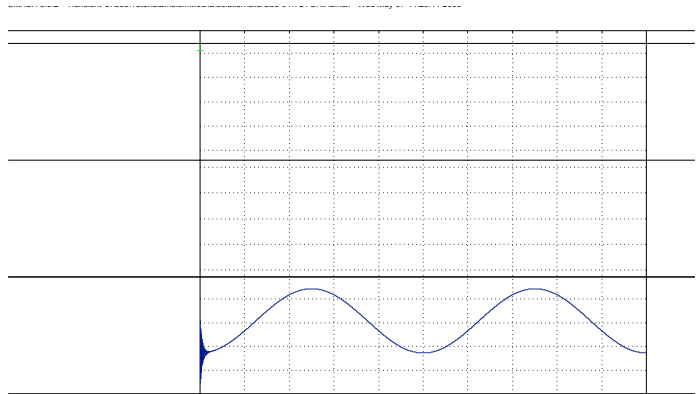
Eine Symbolbibliothek für **Cohesion AMS Designer** ist mit der Modellbibliothek verbunden und unterstützt die Modellierung von kompletten mechatronischen Systemen in seinem Schaltplaneditor. Durch die Möglichkeit der automatischen Netzlistengenerierung kann die Netzliste des gesamten Systems für die Simulation mit **SMASH** erstellt werden.



Schaltplan des mikro-mechanischen Beschleunigungssensors

• **Simulation**

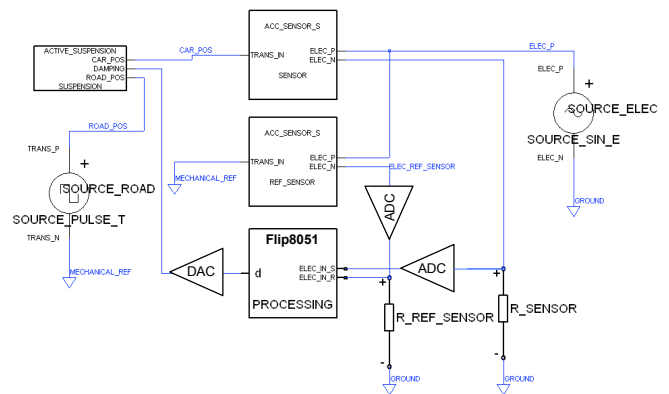
Nach der Generierung einer Testbench, die einen Stimulus für die elektrischen und mechanischen Terminals enthält, kann der Sensor selbst simuliert werden. Die rechte Abbildung zeigt in der ersten Zeile die Position der Masse, in der Zweiten die Kräfte und die Kapazität der Kamms in der letzten Reihe.



**Simulation eines Beschleunigungssensors**

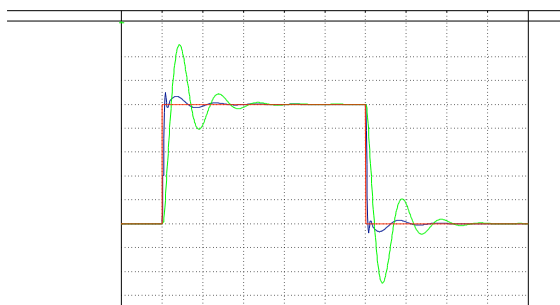
**Makromechanisches System**

Dank des Modells des mikro-mechanischen Sensors kann das gesamte System nun modelliert werden. Das elektrische System besteht aus einem Mikrocontroller (Flip8051), der, mit einem eingebetteten Programm laufend, die Aufhängung des Autos steuert. In Funktion der Echtzeitinformationen, die durch den Sensor bereitgestellt werden, steuert das Programm die aktive Aufhängung, um die Dämpfung zu erhöhen oder zu verkleinern, um die Fahrt komfortabler zu gestalten.



**Modellierung einer semiaktiven Radaufhängung**

Die Beschleunigungssensoren sind über analog/digital Umsetzer (ADU) mit einem FLIP8051 verbunden, während das Feedback der aktiven Aufhängung durch einen digital/analog Umsetzer (DAU) verarbeitet wird.



**Simulation einer halb-aktiven Radaufhängung**

• **Hierarchie**

Die verifizierten und optimierten mikromechanischen Modelle sind bereit, um in ein komplexes System hierarchisch integriert zu werden. Diese Art der Entwicklung verhindert, dass Sie das Rad ständig neu erfinden müssen. In diesem Beispiel einer semiaktiven Radaufhängung wird die einfache Integration der Modelle noch deutlicher.

# Anwendungsbeispiel: Mikrocontroller

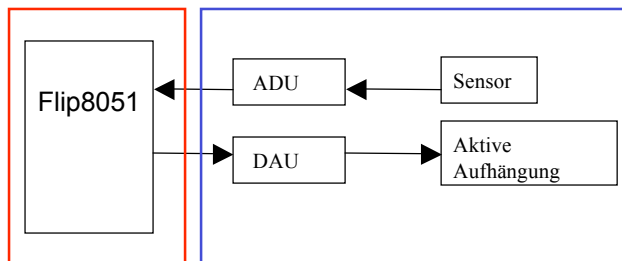
## • Wiederverwendbarkeit

Der Beschleunigungssensor, der aktive Dämpfer und die elektrischen Umsetzer (ADU, DAU) kommen von geeigneten Bibliotheken. Die Modellierung der weiteren Elemente der Radaufhängung wird durch die Wiederverwendung der gleichen Basiseffektmodelle, wie diejenigen, die im Modell des Beschleunigungssensors bereits genutzt werden, erleichtert. Für diese Anwendung müssen nur die Parameter für die mechanischen Basiselemente der Anwendung - der realen Aufhängung – angepasst werden.

## • Auswahl des geeigneten Prozessors für die Anwendung

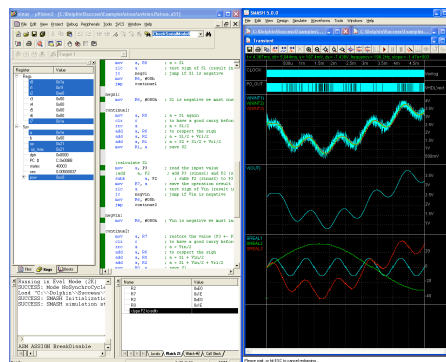
SUCCESS™ ermöglicht es, von dem Multi-Ebenen Simulator SMASH™ im Verbund mit der Integrierten Entwicklungsumgebung von KEIL,  $\mu$ Vision2™, zu profitieren. Das Anwendungsprogramm und der Mikrocontroller (FLIP8051) werden mit  $\mu$ Vision2™ simuliert, während die Peripherien mit SMASH™ simuliert werden. Dank SUCCESS™ wird es ermöglicht, die Verarbeitungsleistung schon zu einem frühen Projektstatus zu bestimmen, die vom System verlangt wird und dann den bestgeeigneten Mikrocontroller für die Anwendung zu finden, was ohne ein solches Werkzeug eine komplexe Aufgabe darstellt.

Die Flip8051 Produktfamilie umfasst verschiedene Produktvarianten von einfacher bis zu elffacher Beschleunigung gegenüber einem Standard 8051er. SUCCESS™ ermöglicht die Auswahl der richtigen Produktvariante für die Anwendung. Dieses Werkzeug versetzt Sie ebenfalls in die Lage zu bestimmen, ob der ausgewählte Prozessor die semiaktive Radaufhängung der vier Räder steuern kann.



Anwendungsprogramm und  $\mu$ P Kern werden im ISS simuliert

Analoge und logische Peripherie werden in SMASH simuliert



## • Validierung der eingebetteten Software

Dank SUCCESS™ wird es schon in einem frühen Projektstatus möglich, die Spezifikation des Systems festzulegen und eine virtuelle Validierung durch Hardware/Software Co-Verifikation durchzuführen.

SUCCESS™ erlaubt es:

- von  $\mu$ Vision2™ IDE zu profitieren
- das Flip8051 Anwendungsprogramm dank der Debugfähigkeiten von  $\mu$ Vision2™ zu entwickeln.
- die Rückkopplung des Mikrocontrollers auf dem mikro-mechanischen System in Echtzeit zu modifizieren und die Simulationsergebnisse direkt zu betrachten.

## IMAGE

Applikationscode kann entweder in einer Assembler- oder in einer C-Sprache entwickelt werden.

Das befehlsgenaue Modell des FLIP8051, simuliert durch den Befehlssatzsimulator (ISS), ermöglicht dem Programmierer, den Anwendungscode für jegliche Peripherie zu entwickeln, sei sie logisch, analog, gemischt logisch/analog oder mechanisch, während sie weiterhin von einer bekannten Entwicklungsumgebung mit Debugfähigkeiten profitieren. Zum Beispiel kann der Entwickler die Reaktion des aktiven Dämpfers unter Verwendung der step by step Funktion des  $\mu$ Vision2™ prüfen, um zwischen “komfortablen” und “sportivem” Modus umzuschalten

- **Schlussfolgerung**

SUCCESS kombiniert die Flexibilität von  $\mu$ Vision2™ IDE, das eine benutzerfreundliche C oder Assembler Quellcode Entwicklungs- und Debugumgebung bereitstellt, mit den Anzeigefähigkeiten des SMASH™ Multi-Ebenen Simulators. Die Simulation des vollständigen multi-Domänen Systems, bestehend aus elektronischen virtuellen Komponenten (analog + logisch), mechanischen Elementen und dem Anwendungsprogramm. Dies erlaubt das Finden von Problemen schon in einem frühen Projektstadium. Dank dieser Co-Verifikationsumgebung sind die Kosten und die Dauer der Entwicklung drastisch reduziert worden.

Ein weiterer Kosteneinsparungseffekt ist die Wiederverwendung vormodellierter und verifizierter Elemente in einer neuen Anwendung.

## Unser Angebot

### • **SUCCESS™**

Die Kombination des wirkungsvollen IDE ( $\mu$ Vision2™) und des Multi-Ebenen Simulators (SMASH™)

- **SMASH™**
  - Simulator für gemischt logisch/analoge Schaltungen
  - multi-Sprachen (C, SPICE, Verilog, VHDL, VHDL-AMS..)
  - multi-Ebenen (Gatter, Transistor, verhaltensorientiert)
    - **EMBLEM-Mecha** - Modellbibliothek von parametrisierbaren (elektro-) mechanischen Effekten
      - universell (implementiert in IEEE standardisiertes VHDL-AMS)
      - verbunden mit Symbolbibliotheken für Schaltplaneditoren
    - **Cohesion Designer** - bietet Ihnen die visuelle Modellierung in einem Schaltplaneditor
      - automatische Netzlistenerstellung Ihres kompletten Systems (elektronische + mechanische Komponenten)
- **$\mu$ Vision2™**
  - integrierte Entwicklungsumgebung (IDE) mit Debugfähigkeiten
    - Assembler
    - Linker
    - C-Compiler
    - Befehlssatzsimulator
  - vollständige Unterstützung der Flip8051 und Flip80251 Features

## Ihre Vorteile

- Modellierung und Simulation Ihres vollständigen Mechatroniksystems
  - einschließlich analoger und logischer Schaltungskomponenten
  - elementare Mechanik (einschließlich Massen, Balken...)
  - elektromechanische Modelle (einschließlich Motoren, Transformatoren...)
  - Softwarekomponenten (HW/SW Co-Verifikation)
  - modulare Entwicklung Ihrer Anwendungen durch leicht wiederverwendbare Modellbibliothek
- Zeit- und Kostenersparnis
  - Design und Verifikation Ihres Konzepts durch Simulation anstatt Prototypenbau
  - effiziente Simulation und Aufdeckung von Problemen bereits zu einem frühen Zeitpunkt der Entwicklung
    - Virtuelle Tests und Optimierungen Ihres vollständigen Mechatroniksystems
    - erhöhen Sie Ihr in-house Wissen durch Training Ihres Personals mittels unserer Trainingsangebote

### **Sie sind an unseren Produkten interessiert und wünschen weitergehende Informationen?**

Zögern Sie bitte nicht uns zu kontaktieren, gerne beantworten wir Ihre Fragen.



Dolphin Integration GmbH  
Bismarckstraße 142a  
D-47057 Duisburg

Telefon: +49 (0)203 / 30622 - 50  
Telefax: +49 (0)203 / 30622 - 69  
email: [mems@dolphin-integration.com](mailto:mems@dolphin-integration.com)

Dolphin Integration SA  
BP65, ZIRST, 39 avenue du Granier  
F-38242 Meylan

Telefon: +33 4 76 41 10 96  
Fax: +33 4 76 90 29 65

Email: [logic@dolphin-integration.com](mailto:logic@dolphin-integration.com)

## Kontakt

Weitere Informationen über SUCCESS™: [http://www.dolphin.fr/medal/success/success\\_overview.html](http://www.dolphin.fr/medal/success/success_overview.html)

Weitere Informationen über MEMS: [http://www.dolphin.fr/flip/mems/mems\\_overview.html](http://www.dolphin.fr/flip/mems/mems_overview.html)

Weitere Informationen über Flip8051 und Flip80251 : [http://www.dolphin.fr/flip/logic/logic\\_overview.html](http://www.dolphin.fr/flip/logic/logic_overview.html)