Chip simulation of automotive ECUs

Ja ob !auss, !att"ias Simons

Abstract

!odern #$%s &ontain ten t"ousands of engine parameters t"at need to be tuned. $alibration of all t"ese parameters is time &onsuming and &omple'. Simulation on a ($ &ould "elp to automate and speed up t"e &alibration pro&ess, in parti&ular if simulation runs mu&" faster)e. g. 20 times* t"an real-time. +o,e-er, engine &alibration is typi&ally performed by an .#!, ,"ile t"e #$% &ode is o,ned by t"e supplier of t"e #$%. /"erefore, t"e .#! is typi&ally unable to set up a #$% simulation based on t"e original $ &ode of t"e #$%. 0nstead, to set up a simulation, time &onsuming and error prone re-erse engineering is needed to de-elop an 1e2ui-alent model1 of t"e #$% fun&tion of interest. /o impro-e t"is situation, ,e "a-e integrated a &"ip simulator into t"e -irtual #$% tool Sil-er. /"is is used to simulate "e' files &ompiled for /ri$ore targets dire&tly on ($. Simulation re2uires

1. a "e' file t"at &ontains program &ode and parameters of t"e simulated fun&tions
2. start addresses of t"e fun&tions to be simulated
3. an ASA(24a2l file t"at defines t"e &on-ersion rules for t"e in-ol-ed inputs, outputs, and &"ara&teristi&s, as ,ell as &orresponding addresses

/"e start addresses of fun&tions &an e. g. be e'tra&ted from a map file generated toget"er ,it" t"e "e' file. Sil-er uses t"e a2l file to automati&ally &on-ert s&aled integer -alues to p"ysi&al -alues and -i&e -ersa during simulation. A /ri$ore simulation &an also be e'ported as S5un&tion)me',32 file* for use in !A/6AB4Simulin . .n a standard ($, "e' simulation runs ,it" about 70 !0(S. 0f only simulating sele&ted fun&tions of an #$%, t"is is fast enoug" to run a simulation mu&" faster t"an real-time. 0n t"is paper, ,e also report "o, su&" simulations are used today to support t"e de-elopment of gasoline engines at 8aimler.

Introduction: Virtual ECUs in the development process

Simulation "as great potential to impro-e t"e de-elopment pro&ess for #$%s. Simulation "elps to mo-e de-elopment tas s to ($, ,pms "s s !2 oteyo teyo

- on a ($, a &alibration tool li e 0;$A)#/AS* or $A;ape)<e&tor* &an be &onne&ted to a -irtual #$% -ia =$(to measure into a running simulation and to tune &"ara&teristi&s online. /"is ,ay, many parameters of a #$% &an be tuned using a relati-ely &"eap and "ig"ly a-ailable ($ platform, ,it"out

models &an be imported from many simulation tools into Sil-er, in&luding !A/6AB4Simulin , 8ymola, Simulation= and !apleSim, e.g. t"roug" t"e 5!0 format for model e'&"ange A7B.

+o,e-er, sometimes $ &ode is not a-ailable for implementing a -irtual #$%. /"ere are t,o main sour&es for su&" a situation:

- Protection of intellectual property: All or maEor parts of t"e #$% "a-e been de-eloped by a supplier and t"e .#! interested in building a -irtual #$%)e.g. to support &alibration, a tas typi&ally performed by an .#!* "as t"erefore no a&&ess to t"e $ &ode.
-

/"e tas s of &ategories 1 and 3 typi&ally depend on details of t"e parti&ular &"ip, and on t"e #$% "ard,are. 0n &ontrast, tas s of &ategory 2 are fairly independent from su&" "ard,are-spe&ifi& details. /o simulate #$% &ode, it is t"erefore &on-enient to run only tas s of &ategory 2. /"e re2uired inputs for t"ese tas s &an eit"er be ta en from measurement files)open-loop simulation*, or t"ey are &omputed online by some plant model)&losed-loop simulation*, bypassing t"e tas s of &ategory 1. 6i e,ise, t"e outputs of &ategory 2 tas s &an be dire&tly &ompared to measurements)open loop* or fed into t"e plant model)&losed loop*, bypassing t"e &ategory 3 tas s. /"e signal interfa&e bet,een &ategories 1-2 and 2-3 is typi&ally ,ell do&umented and a-ailable, e.g. from t"e $A; spe&ifi&ation)8B$ file* of t"e #$%.

/"is modelling strategy "as a -ery good &ost-benefit ratio. 0n order to simulate also t"e tas s of &ategories 1 and 3, one "as to model "undreds or perip"eral and &"ip spe&ifi& registers, and to build state-ma&"ine models for lo,-le-el perip"erals, su&" as $A; &ontrollers. /e&"ni&ally, t"is is possible, e. g. ,it" System$ A?B, but "ardly Eustified by t"e added -alue, at least for t"e appli&ations &onsidered "ere.

Sil-er 2.? uses a spe&ifi&ation file)similar to t"e .06 file used to &onfigure .S#H* to spe&ify, ,"i&" tas s of a "e' file to simulate. Sil-er automati&ally turns su&" a spe& file into an e'e&utable Sil-er module)dll* or S5un&tion. A typi&al spe& file loo s as follo,s:

```
01 # specification of sfunction or Silver module
02 hex_file(m12345.hex,  ri!ore_1.3.1"
03 a2l_file(m12345.a2l"
04 map_file(m12345.map"     # a  #S$%&' or '&( map file
05 frame_file(frame.s"       # assem)ler code to emulate * +S
0, frame_set(S -._S%/-, 10"  # Silver step si0e in ms
01 frame_set( -2 _S #* , 0xa0000000" # location of frame code
03
04 # functions to )e simulated, in order of execution
10 tas5_initial(#6!7-_ini"
11 tas5_initial(#6!7-_inis8n"
12 tas5_tri99ered(#6!7-_s8n, tri99er_#6!7-_s8n"
13 tas5_periodic(#6!7-_20ms, 20, 0"
14 tas5_periodic(#6!7-_200ms, 200, 0"
15
1, # interface of the 9enerated sfunction or Silver module
11 a2l_function_inputs(#6!7-"
13 a2l_function_outputs(#6!7-"
14 a2l_function_parameters_defined(#6!7-"
```

/"e "as" # &"ara&ter starts a &omment, ,"i&" is ignored by Sil-er. /"e spe& file first lists t"e re2uired files)line 2-?*. /"e map file is optional. 0f a map file is gi-en, t"e spe& file may use symboli& names f@ fssBő emulation. 5or e-ent triggered tas s, Sil-er offers t,o alternati-e e-ent models. 6ine 12 s"o,s a fun&tion t"at is e'e&uted n times at ea&" Sil-er step, ,"ere n is t"e -alue of t"e input -ariable tri99er_#6!7-_s8n at t"e beginning of t"e step. /ypi&ally, n is 0 or 1 during simulation. +ig"er -alues o&&ur only, ,"en more t"an one trigger e-ent o&&urs during one step. Sil-er also offers a more a&&urate e-ent model, t"at allo,s e'e&ution of an e-ent triggered tas at e'a&t e-ent time, not Eust at t"e beginning of a step.

5inally, lines 1G-19 define t"e inputs, outputs and parameters of t"e generated module or S5un&tion. 0n t"is &ase, ,e Eust reuse t"e interfa&e of a 5%;$/0.; element of t"e a2l file, for a fun&tion &alled AB$8#. 0t is also possible, to list indi-idual -ariables "ere by name, as long as t"eir properties)su&" as address, &on-ersion rule, data type* are des&ribed in t"e a2l file.

0n addition, t"e spe& file offers means to spe&ify

- properties of t"e =$(emulation, if any, to support online &alibration and measurement using tools su&" as 0;$A and $A;ape
- data se&tions to be in&luded into t"e generated Sil-er module or S5un&tion. /"is ,ay, initial loading of t"e "e' file into simulated memory &an be a-oided, to speed up simulation.
- memory areas to be &opied to ot"er)faster* memory by t"e start-up &ode
- fun&tions to be repla&ed by ot"er fun&tions. /"is ,ay, a fun&tion &alled by a tas of &ategory 1 or 3 to a&&ess sensors or a&tuators &an be repla&ed by a fun&tion t"at dire&tly a&&esses a plant model or measured -alues instead.
- logging options, e.g. to tra& memory a&&ess during simulation

/"e Sil-er module or S5un&tion generated t"is ,ay performs e'a&tly t"e same &omputations on ($, as on t"e real target, sin&e t"e effe&t of e-ery ma&"ine instru&tion on memory and &"ip registers is e'a&tly simulated on ($. +o,e-er:

- simulation is Eust instru&tion a&&urate, not &y&le a&&urate. /"is means, t"e simulation on ($ &annot be used to e'a&tly predi&t e'e&ution time on t"e real target. 5or e'ample, pipeline effe&ts of different a&&ess times to memory)e.g. fast on-&"ip CA! -s. e'ternal CA!* are not modelled.
- &on&eptually, simulated tas s e'e&ute infinitely fast. /"is means t"at t"e emulated C/.S ne-er interrupts a tas . /"e &orresponding effe&ts &annot be analysed using t"e generated model.
- Sili&on bugs are not simulated. 0f a &ompiler for t"e real target does not ,or

instru&tion. 0t is also possible to set &ode and data brea points, for e'ample to pause a simulation ,"ene-er a &ertain -ariable is a&&essed.

0n order to measure t"e e'e&ution speed of &"ip simulation, ,e "a-e ported a &omple' #$% fun&tion implemented by ? different $ fun&tions t"at run initially, e-ery 10 and 200 ms, and syn&"ronous to t"e &ran s"aft. /"e spe& file is -ery similar to

! "#$ %

Sil-er &an also turn a spe& file as des&ribed in se&tion 2.1 into a S5un&tion, i.e. a me',32 file t"at runs in Simulin . /"is is parti&ularly interesting ,"en using &"ip simulation to support automated optimi9ation of parameters, be&ause many optimi9ation tools are implemented on top of !A/6AB4Simulin . /"e generated S5un&tion a&&epts all &"ara&teristi&s listed in t"e spe& file as S5un&tion parameters. /"is ma es it easy to &onne&t t"e generated S5un&tion ,it" an optimi9ation pro&edure. 5or e'ample, t"e S5un&tion &an be &alled ,it" ,or spa&e -ariables t"at are t"en automati&ally -aried by t"e optimi9ation pro&edure bet,een S5un&tion &alls. /"e performan&e of a generated S5un&tion is again about 70 !0(S.

Applications of chip simulation

0n t"is se&tion, ,e s"ortly s et&" &urrent appli&ations of t"e presented approa&" at 8aimler.

"& % ' ( ) *

8uring de-elopment of an engine &ontroller, a de-eloper mig"t ,ant to repla&e a &ertain fun&tion of t"e #$% by its o,n -ersion of t"at fun&tion, bypassing t"e original fun&tion. 5or real #$%s, t"is &an be done ,it" tools su&" as #+. . HS)#/AS* or ;o-+oo s)A/0*. /"ese tools manipulate t"e original "e' file, su&" t"at t"e bypassed fun&tion is not e'e&uted any more, but Eust &alls t"e ne, fun&tion instead. /"e ne, fun&tion is e. g. de-eloped ,it" !A/6AB4Simulin in &onEun&tion ,it" a &ode generator and a &ompiler for t"e target pro&essor. /"is met"odology still re2uires a&&ess to t"e real #$%: t"e manipulated "e' file needs to be flas"ed into t"e #$%, and t"e #$% needs to run t"e ne, fun&tion, su&" t"at its be"a-iour &an be assessed. 0n order to furt"er simplify t"e assessment of t"e ne, fun&tion, ,e e'e&ute t"e manipulated "e' file in Sil-er on ($ using &"ip simulation as des&ribed abo-e. Su&" simulations are typi&ally dri-en open loop by measurement files)!85*.

/"e pla&ing of bypass "oo s by dire&t manipulation of t"e "e' file is a mig"ty but error-prone tool. Sometimes a "oo ed fun&tion is not &alled at all or only some -ariables are o-er,ritten and some not. ;ormally, su&" errors are only dete&ted after

, it" !A/6AB4Simulin . /"is "as been time &onsuming and error prone. >e "a-e no , partially repla&ed t"ese "and-&oded models , it" S5un&tions generated automati&ally by Sil-er from a gi-en "e' file. /"e generated S5un&tions proofed to run as fast as t"eir "and &oded &ounterparts. /"e repla&ement of "and-&oded floating-point models by generated fi'-point S5un&tions raises t"e follo , ing problem: Some optimi9ation pro&edures re2uire gradient information to guide t"e sear&" for optimal parameter -alues. 5or e'ample, , "en sear&"ing for an *)* t"at minimi9es f)*)*), t"e deri-ati-e *!*f,*!)* is to be &omputed during optimi9ation for different -alues of *)*. 5inite differen&es are often used "ere, i.e. *!*f,*!)* is &omputed as)f)*)* I #* - f)*)*** 4 # for small #, say # J 10^{-D}. 0f f is &omputed using &"ip simulation, *)* and *)*-# are often bot" mapped to t"e same integer, resulting in a 9ero gradient. As a &onse2uen&e, t"e optimi9ation pro&edure is la& ing guidan&e, and mig"t return a suboptimal solution. /"ere are also so-&alled deri-ati-e-free pro&edures for optimi9ation. . b-iously, t"ese are not affe&ted by t"e abo-e problem.

Conclusions

As demonstrated abo-e, an #$% "e' file &ompiled for some target pro&essor &an be e'e&uted by t"e -irtual #$% tool Sil-er on >indo , s ($, eit"er open-loop dri-en by measurements or in &losed-loop , it" a -e"i&le model. 8epending on t"e appli&ation, sele&ted #$% fun&tions are simulated, or nearly t"e entire #$%.

/"is ind of simulation opens ne , possibilities to mo-e de-elopment tas s from road, test rig or +i6 to ($s, , "ere t"ey &an be pro&essed faster, &"eaper or better in some respe&t, , it"out re2uiring a&&ess to t"e underlying $ &ode. 8aimler &urrently uses t"is inno-ati-e simulation approa&" to support &ontrols de-elopment for gasoline engines. . t"er appli&ations, su&" as online &alibration on ($ -ia =$(seem to be doable as , ell.

!eferences

A1B A. Jung"anns, C. Ser , ay, /. 6iebe9eit, ! . Bonin: Building <irtual #$%s @ui& ly and #&onomi&ally, A/K ele troni 0342012, Juni 2012. See , , , .A/Konline.de or "ttp:442troni&.de4do&4A/KeL2012Len.pdf

A2B +. Brü& mann, J. Stren ert, %. Heller, B. >iesner, A. Jung"anns: ! odel-based 8e-elopment of a 8ual-$lut&" /ransmission using Capid (rototyping and Si6. 0nternational <80 $ongress /ransmissions in <e"i&les 2009, 5riedri&"s"afen, : ermany, 30.0D.-01-0G.2009. "ttp:442troni&.de4do&48$/L2009.pdf

A3B H. CMp e)ed.*: 8esign of #'periments)8o#* in #ngine 8e-elopment - 0nno-ati-e 8e-elopment ! et"ods for <e"i&le #ngines. #'pert <erlag, 2011.

A7B /. Blo&" , it9, ! . . tter et. al.: 5un&tional ! o& up 0nterfa&e 2.0: /"e Standard for /ool independent #'&"ange of Simulation ! odels. 9t" 0nternational ! odeli&a $onferen&e, ! uni&", 2012.

A?B System$, 6anguage for System-6e-el ! odeling, 8esign and <erifi&ation, see , , , .system&.org

ADB ! . /atar, C. S&"ai&", /. Breitinger: Automated test of t"e A ! : Speeds"ift 8$/ &ontrol soft , are. 9t" 0nternational $/0 Symposium 0nno-ati-e Automoti-e /rans-missions, Berlin, 2010. "ttp:442troni&.de4do&4/est>ea-erL$/0L2010Lpaper.pdf

•

. r/ *0*ako*"* *(* auss+ *1* Tronic *%*m*"2*+ *3*lt- *(* oa*"*it *4**+ 1*5664* *$*erlin
(att#ias Simons+ *.* aimler *3%*+ G0?7D Stuttgart