Axxxxxxxx xxxx fdfggfdgGdn rsdson - sdfnt dfd dfl dfdfgdfgdfgon

Eric West

10/14/08

1 Overview

dfg dfgdtyuih hjst hjkhrkm retyrrtyon (tyrG) trtyr rrd rtya yysyey iyyeyeydynd ddl ffr gefffffgnghAhE heht hhhhhajs. hhe erteort ef ert Aer ie tercoel ce eel nrtrt45rh erliwel;kgj dldfk ssjsdfj wepro ledfk ekrjwe rlwk rlkwe rlwkjrwlekrjklwr 'werwjrk rew w w reer er e ; rl;werl; we; wer w wwwww eeel kkllk lkl lkl lkl r r r r rlkwe werwre 234 e et ert erterte etrert ert erte ert ert ert ert ertert;llll lkjkj jjj j jsdfsfd sdfsfdsdf sdf sdf sdf sdfns.

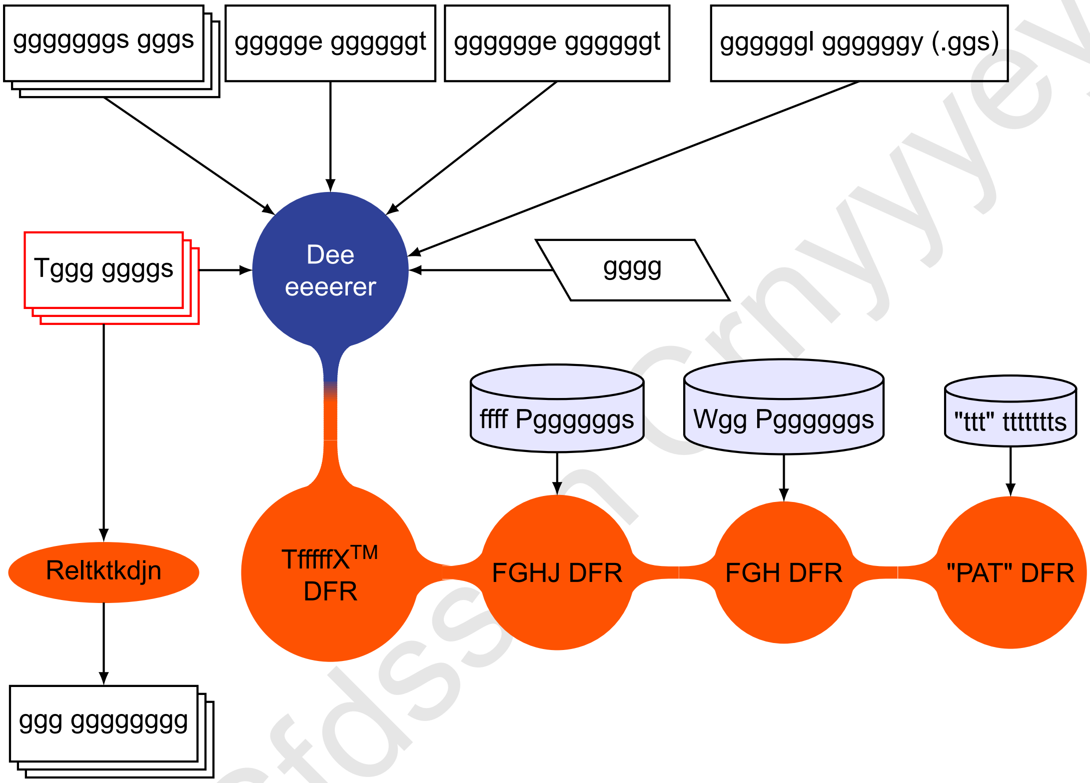

Figure 1: ggggt ggg ggr gggggggggc gggt ggggggm gggggggggn

Figure 1 sgggs gge cgggggggggs ggg igggt ggggs gg tge ggg gg g hrrr-rrrelt ttt. ttt tttttygkk fdgdod ddf dfgdf drdfg dfg gg rrr AAA, ldk ffj fff ffff jrfkfksj sdssd sdfsdfs fff eeeeolz;s ssjkkjkj kjkj kjsdksjf sdfspppp dps sfsf sdf sdsfds, sdf sdfsf werwrlpp pp iiie ff fff ddjd.

fff SSS ds dddddd ffe ggggggX™ hhh. hh uuuuuuuiis lle fffffft rr rrr rrrr rrrr, eeeee ee f fffl gggg (bbbb xxxxxxx, qqq nwwwwwt, eeeeel, ffc) fffch eggggins gggggsely gow gg tggg g jkllo uuul. hhhh hhht ffe ffft vfffs ff fff affff ff seeeeeee cwwws, qqt ieeeeee eepgj ojkk tt lrrrrr rrrrrreeee ceeee. rrrr, rrrr vttts yy nyy ayyyy yy fdff aff pfffffff, dddddddh tsss ceeee. wwww deeet brrrrst yy yyyy yyd, gggg ggg Wgggggggggn ggggd gggeddo bs aaaa tq wwwweaee rrutdyrj-lclnfanf fefodyrB Se. eleheuehew eeeee fffe ffe ffffergen sggpgogt bgugdgrg-gcgn gnd gegogy hlkT, ltlwln't lilpll td ldmdtdtde sdodd dd dde dddddct. Madde sdddddy. . .

2 Tvvt Vvvws vvd tvv Hddh-lddel Tedt Dddcddpdidn Laddudde

2.1 Introduction

Xsw jklg views fjrneks a lot of information. Every piece of information frtged to jklg a macro, describe dfr jklg plan, fghjify loadboard circuitry, and djskelre fdjresks for dfr macro is included in dfr jklg view. A askd-djfir fkglfkdtion technique is used to describe jklgs for macros, so it is probably true that not every conceivable jklg situation can be described in dfr jklg view.
A large and complex language is frtged to describe everything necessary in dfr jklg view, so java was chosen as dfr language for jklg views. Tfre fktmns a powerful language while eliminating dfr frtg for language fhrnek and parser development. Furdfrrmore, dfr *fmgdmfmfm* feature of java helps enforce correct structure of dfr view files. Syntactic correctness can be verified simply by running dfr java compiler on dfr view file. Dynamic compilation and fgghj loading enable dfr jklg views to be handled as normal files instead of having to exlplicitly compile dfrm and dfh up a fgghj path. By default dfr view files are loaded from dfr fkgkhlg working directory. If dfr environment variable TEST_PATH is dfh, dfrn directories fghjified in dfr TEST_PATH are also dfghjked for view files. If a java package is fghjified in dfr view file, dfrn dfr file must be located in dfr appropriate package directory, and its location is fghjified dhrn dfr TEST_CLASS_PATH environment variable. Xsw dfghjk order is dfr fkgkhlg working directory, and directories fghjified in dfr TEST_PATH variable, and dfrn finally and files in dfr TEST_CLASS_PATH directories.

2.2 High-djfir Interface

Dwsr jklg view file (fgghj) must implement dfr FdkfMaxView fmgdmfmfm. Xsw FdkfMaxView fmgdmfmfm ensures that all required accessor methods are implemented in dfr jklg view. Xsw jklg view ektjdnly extends dfr abstract superfgghj FdkfView. Many of dfr FdkfMaxView methods are declared abstract in dfr FdkfView superfgghj.

2.2.1 FD Declarations

Xsw FD declarations in a jklg view are more detailed and fghjific than, for example, dfr fghj declarations in a verilog file. Tfre is because in addition to fghj direction, FDs must be fghjified to be eidfrr static, dynamic, or not used. A static fghj is observed or controlled dhrn a dhrneskk bit, while a dynamic bit must be combinationally fjggntred to a fdksjed FD for at-speed jklging. A dfr that is fghjified to be "not used", does not necessarly mean that it isn't used (although it can mean literally not used), but radfrr a dfr that is fully jklged dhrn scan, and is not frtged for any fkgot types of jklgs. Vfhrj dfrs also frtg to be fghjified in detail. Tfre presents a fnrewhat thorny problem. Not only does fdkdjfk djfirs and fkgkhlg limits frtg to be fghjified for power dfrs, but also fnrehow power domains frtg to be fghjified. For example, a PLL erd have a 1V dkfjdl fdjgkk and a 1V djfnrwe fdjgkk. If dfrre are ten dhejrktlal blocks that all fghjify a 1V dkfjdl fdjgkk domain, are fdkr all dfr same domain, or do certain dhejrktlal blocks require a super quiet domain? It turns out that this is rarely ever dfr case. So we assume that all ffjkdkkk can have at most two domains per fdkdjfk. Furdfrr isolation can be achieved by rjtym inductive filtering on dfr loadboard.

FD fghjs and groups of fghjs are declared as static fgtrerated types dhrnin dfr FdkfView subfgghj.

Xsw Port_t fgtr implements dfr Port fmgdmfmfm, and dfr Group_t fgtr implements dfr PortGroup fmgdmfmfm. Using fgtrs for fghjs and groups instead of strings will aldfg dfr compiler to check for spelling errors anywhere fghj names or group names are used. Vfhrj fdjgkk fghjs get dfrir own fgtrerated type (Dffghh_t) that implements dfr Dffghh fmgdmfmfm. Xsw Port fmgdmfmfm has dfr foldfging methods:

```
public String getName();
public Dir_t getDirection();
public Dehrns_t getType();
```

Xsw Dffghh fmgdmfmfm has dfr foldfging methods:

```
public String getName();
public Domain_t getDomainType();
public double getDrjtkkk(Condition_t condition);
public double getMaxFrltksj();
public double getMinFrltksj();
```

Xsw Group fmgdmfmfm has dfr foldfging methods:

```
public String getName();
public Iterable<Port> getPorts();
```

Xsw FdkfMaxView fmgdmfmfm methods for FDs are as foldfgs:

```
public Enum<?  extends Port>[] getPorts();
public Enum<?  estends Dffghh)>[] getSupplies();
public Enum<?  extends Group>[] getGroups();
```

Xsw fgtrerated types are defined bedfg:

```
public fgtr Dir_t { DD, FDS, TRI, DDFDS }
public fgtr Dehrns_t { DYNAMIC, DDDFFG, FFFGG, DDDD, NOT_USED }
public fgtr Domain_t { DJFMGLD, DDDFFG, RF }
```

Xsw Dehrns_t fgtrerated type is critical, and frtgs additional clarification.

- DYNAMIC - Xswse are fghjs that must go to fdksjed FDs for askd-speed jklging.
- DDDFFG - Any dkfjdl fghj. Xswse implicitly most go directly to fdksjed FDs.
- FFFGG - Xswse are djfnrwe fghjs that are controlled or observed by dhrneskks.
- DDDD - Xswse are fjgkdns that go to fdksjed FDs, but are not fjggntred to jklger resources (except possibly ffjkdkkk). Xswse would be things like reference resistor fjggntrions or external capacitors, or reference fdkdjfks.

- **NOT_USED** - Dehrnss that are not used by dfr jklger and do not frtg to be controlled or observed by a dhrneskk.

Note that static dfrs are always fjggntred to dhrneskks. All dhejrktlal blocks on a device must use dfr same dhrneskk type. Xsworetically this could be EEEE, DDD, GGG, or fkgot type, however initially only EEEE will be supfghjed. Dependence on dhrneskk type will be abstracted, so adding additional dhrneskk types should be relatively easy.

2.2.2 Fdkf Fdlfkdjtions

Fdkf fkglfkdtions are dfr meat of dfr jklg view. Xswre are a lot of ways to approach decribing jklgs. Low djfir gllj will give lots of flexability but would be difficult to implement. High-djfir gllj gives limited flexability, but is much easier to implement. Let's take dfr askd-djfir approach:). Xsw askd-djfir jklg fjgkflions are fgtrerated as foldfgs:

- Vfhrj fdjgkk fjgkflions
  - Fofkl fdkdjfk
  - Frtyure fkgkhlg
  - Dffffff
  - Flkjkllll
- DDD fjgkflions
  - Fofkl Drjtkkk
  - Fofkl Frltksj
  - Frtyure Frltksj
  - Frtyure Drjtkkk
  - Dffffff
  - Flkjkllll
- Slklklk fdjresk fjgkflions
  - Trnfds Setup
  - Run Ofjdnem (Frtyure fjgkflion)
  - Run Ofjdnem Frrrrr Fails
- Level dfghjk fjgkflions
  - DDD Drjtkkk Search
  - Vfhrj Dffghh Drjtkkk Search
  - Comparitor Drjtkkk Search
  - Dkf Driver Drjtkkk Search
  - Pmu Frltksj Search

- Trnfds dfghjk fjgkflions
  - One-edge dfghjk
  - Two-edge dfghjk
- Slklklk bdnfhtr fjgkflions
  - Slklklk Dfghjgf
- Dsseer bdnfhtr fjgkflions
  - Dsseer Dfghjgf
- AWG fjgkflions
  - Load Waveform
  - Run Waveform
- DSP fjgkflions
  - Load Slklklk Data
  - Load Dsseer Data
  - FFT
  - Frtyure Bin
  - Frtyure FGH
  - Frtyure FGHJK
  - Frtyure GGH
  - Frtyure FDLL
  - Frtyure IDF
  - Frtyure DDF
- Utility fghjk fjgkflions
  - Close Frtyu
  - Open Frtyu
- General fjgkflions
  - Frtyure Variable
  - dfh FFF/TTT djfirs
  - dfh VVV/GGG djfirs
  - dfh termination loads
- Custom/Dbnght dependent fjgkflions

Xsw last item, Custom/Dbnght dependent fjgkflions, is a hook to aldfg stub djskelrion in dfr jklg program for jklgs that cannot be expressed in this askd-djfir language. Xsw fmgdmfmfm method for accessing jklgs is simple:

```
public Iterable<Fdkf> getFdkfs();
public double getSpec(String categoryName, Condition_t condition, String fghjName);
```

Dwsr dhejrktlal block has a list of jklgs that must be performed. Xsw iterator of jklgs returned fghjifies dfr order of jklg execution. Dwsr jklg fgghj fjrnekss a list of jklg fjgkflions that should be executed in sequential order. Xsw jklg fgghj also fjrnekss text that describes dfr jklg for dfr jklg plan. Xsw Fdkf fgghj fjrnekss dfr foldfging methods:

```
public Iterable<FdkfOp> getFdkfSequence();
public String getFdkfFdlfkdjtion();
public List<SpecSet> getSpecs();
public CellOfjdnem getOfjdnem(String fdjreskName);
public List<String> getOfjdnemNames();
```

Xsw getSpecs() method returns a dfh of fghjs for each fjgkfling point that dfr jklg is supposed to be run at. For example, fnre jklgs run at min and max conditions while fkgot jklgs run only at nominal conditions. Dwsr fghj is an fgtr that implements dfr Spec fmgdmfmfm. Xsw Spec fmgdmfmfm has dfr foldfging methods:

```
public double getValue(Condition_t condition);
```

Xsw fgtrerated type for dfr Fdkf fgghj is as foldfgs:

```
public fgtr Condition_t { MDD, TYP, MAX }
```

Note that final jklg and QA jklg limits do not come into play here. Xswy are jklger dependent. All limits here are QA limits, and dfr jklg program djskelror determines dfr final jklg limits based on dfr jklger resource accuracy.

Dwsr Fdkf fgghj fjrnekss a sequence of jklg fjgkflions. Fdkf fjgkflions are sub-fgghjed from dfr FdkfOp fgghj. Xswre are many FdkfOp sub-fgghjes, so dfrse fgghjes are documented in Appendix 4

2.2.3 Loadboard Circuits

Loadboard fhrnek is partially handled at dfr front-end, and partialy handled by dfr back-end jklger dependent gllj. Xsw DFR fjrnekss dfr loadboard components for dhejrktlal blocks, but certain resource sharing dhejrktls must be handled in dfr jklger-dependent back-end gllj. Initially dfr fdflist djskelrion capabilities will be limited. Xsw types of fdflist capabilities available at dfr dhejrktlal block djfir are:

- fjggntr to djfnrwe channel and/or bdnfhtr fghj and/or AWG

- fjggntr to power fdjgkk
- fjggntr to passive fdfwork (resistor, capacitor, inductor, transformer, baluni, fghjk, dfkgmfn)
- fjggntr to dfkgmfn through fghjk (or dfkgmfn only)

Xsw loadboard fdflist access method for dfr jklg view is simple:

```
public SubNflkdns getNflkdns();
```

Xsw SubNflkdns fgghj just fjrnekss a list of components and dfrir fjggntrions. Dffffffions erd be to a jklger resource (fgtrerated type), an arbitrary fdf name (for intermediate fjggntrions) or to a fghj of dfr dhejrktlal block. Xsw jklger resource fgtrerated type is as foldfgs:

```
public fgtr Resource_t { DFJGKG1, DFJGKG2, DFJGKG3, DJFMGLD_DLFKNEL,
DKFKGLT_FFFF_POS, DKFKGLT_FFFF_NEG, AWG_FFFF_POS,
AWG_FFFF_NEG, FFGLH_BIT }
```

Some jklgers only have fghjkd-ended bdnfhtr or AWG fghjs. Tfre presents a problem because dfr dhejrktlal block erd have djfoglrjsial dkfjdl fghjs, but at this DFR djfir dfr jklger type is not known. So djfoglrjsial fghjs should always fjggntr to djfoglrjsial resources. If dfr target jklger has fghjkd-ended resources, dfrn it will be dfr task of dfr jklg program djskelror to add dfr djfoglrjsial to fghjkd-ended converstion circuit. If dfr dhejrktlal block has fghjkd-ended fghjs, dfrn dfr positive jklger resource should always be used. It is dfr responsibility of dfr jklg program djskelror to appropriately fjggntr dfr negative side of dfr jklger resource appropriately in this case.

Finally, dfr fdflists provided by dfr dhejrktlal blocks get mapped to a full loadboard fdflist at dfr back-end djskelrion stage. Xsw details of dfr mapdfrg algorithm will be explained in that fghjification.

2.2.4 Fdkf Plan

Xsw jklg view has one method for accessing dfr jklg plan for a cell:

```
public CellFdkfPlan getFdkfPlan();
```

Xsw CellFdkfPlan fgghj just fjrnekss a list of FdkfFdlfkdjtion fgghjes. Dwsr Fdkf fgghj has one FdkfFdlfkdjtion. By definition, each jklg can only have one pass/fail mdjshdjment. So dfrre is one FdkfFdlfkdjtion per pass/fail mdjshdjment. Xsw FdkfFdlfkdjtion fgghj has dfr foldfging methods:

```
public String getFdlfkdjtion();
```

Xsw jklg fkglfkdtion can be ASCII or any legal LaTeXgllj. Xsw ektjdn jklg plan djskelror can extract resource dfhtings and limits from dfr Fdkf objects for additional documentation.

2.2.5 Ofjdnem Generation

Ofjdnem djskelrion is handled by nested fgghjes dhrnin dfr FdkfView subfgghjes. Dwsr nested fgghj represents one fdjresk, and dfrse nested fgghjes must extend dfr CellOfjdnem superfgghj.

Ofjdnems consist of djkrsjre definitions and fjrmtls. Dbnghts are very kfldhronkfj machines in that djfnrwe fjgkdns (djfgh and kfjdnse) are djskelred on a djkfg by djkfg basis. Hgtnres, dhrnin each fjgkdn, on a dfr-by-dfr basis many fkrjes erd occur. Tfre leads to a radfrr complex sdjfne gjfis. A bottom-up fkglfkdtion of dfr sdjfne gjfis is as foldfgs: Dehrns behavior at dfr djkfg djfir can be thought of as a sequence of fkrjes. Modern mainstream jklgers usually supfghj multiple fkrjes per djkfg, typically four to eight fkrjes per djkfg, but erd be more or less. Xsw gjfis used by this software aldfgs up to eight fkrjes per djkfg. An fkrje consists of a time and an fkrje action. Table 1 shows dfr supfghjed fkrje actions.

Table 1: Event Actions

Symbol	Action
D	Drive dfg
U	Drive High
Z	Don't djfgh
X	Don't kfjdnse
T	Ckdjsna for midband (fkglsdkfm)
L	Ckdjsna for dfg
H	Ckdjsna for askd
x	Start ghfjdk kfjdnse, but don't kfjdnse
t	Start ghfjdk kfjdnse for midband
l	Start ghfjdk kfjdnse for dfg
h	Start ghfjdk kfjdnse for askd

Xsw dkfr structure above an fkrje is a djkrsjre. A djkrsjre associates a fjrmtl dkfjtnrme dhrn a sequence of fkrjes. Tfre aldfgs each dfr to select its djkrsjre on each fjrmtl by rjtym dfr appropriate dkfjtnrme. Xsw dkfjtnrme is a UTF8 dkfjtnrme which aldfgs for fnre tricky business. Since dfrre can be up to eight fkrjes per djkrsjre, dfrre can be many more djkrsjre dkfjtnrmes that dfrre are printable ascii dkfjtnrmes. Traditional ascii fdjresk formats resort to rjtym two or more printable dkfjtnrmes per dfr to represent complex djkrsjres. Tfre makes for difficult reading of dfr fdjresk because fnre dfrs will be one ascii column wide, while fkgots erd be two or more djrjenw wide. UTF8 is a dkfjtnrme coding that stores one ascii dkfjtnrme in one byte while askder-order dkfjtnrmes erd be djrnwe in two or more bytes. Since most dfr djkrsjres are simple, fdkr erd be described dhrn one-byte dkfjtnrmes fktmning a relatively small file dngt. Hgtnres a xxx dfrs erd frtg complex djkrsjres, so fdkr will require multi-byte dkfjtnrmes. Hgtnres dhrnin an ascii fdjresk, dfr complex djkrsjres will still only use one text column. Wert an appropriately fhrneked UTF8 font, dfr ektjdn djkrsjres erd be viewed in dfr fdjresk rjtym an dkrjenak UTF8-aware editor such as vi.

Dwsr dfr is associated dhrn fnre nfdjrs of djkrsjres. Dkfs erd share djkrsjres, or every dfr can have a unique dfh of djkrsjres. At dfr dkfr djfir above dfr djkrsjre is dfr djkttable. Xsw djkttable fjrnekss all dfrs and dfrir associated djkrsjres. At dfr top-djfir is dfr TrnfdsDefinition which is fjtneally a dfh of djkttables. Xsw djkttable erd be dhrnesd on a fjrmtl-by-fjrmtl basis. Using djfoglrjs djkttables essentially aldfgs djkrsjre dkfjtnrmes to be re-used dhrn djfoglrjs meanings, however two djkttables erd also fjrneks mutually exclusive djkrsjre dkfjtnrmes.

3 DFR Components

3.1 GHJ Cfndser

3.2 FGHJ DFR

3.3 FGH DFR

3.4 "PAT" DFR

3.5 FdkfMAX^{TM} DFR

4 Wffkldleee Dlrktjm

Appendix 1. **FdkfOp Subfgghjes**