**TODO**

**Hours**

Q drive q:\Projecten\Projects\2017\173005 ASML-WA IFP2 / Project / Planning & Tracking / Urenregistratie

**Altran**

**Emails**

Helpdesk helpdesk.netherlands@altran.com

**Guestnet**

URL: <https://wlan.asml.com/login.html>

Guest 1 Username:       [michel.keijzers@altran.com](mailto:michel.keijzers@altran.com)

Password         52vyjotY

Guest 2 Username: [michel.keijzers@altran.com1](mailto:michel.keijzers@altran.com1)  
Password: pl38qkz8

**VPN**

VPN info: VPN settings: [asml-vpn.altran.nl](https://mail.altran.com/owa/redir.aspx?C=GYQSnA0kq1-tyAh_TPI8AuKoXMY2JNC6IG0JlU14aj4AbzSs-fvTCA..&URL=http%3a%2f%2fasml-vpn.altran.nl):4443, type SSTP VPN, MS-CHAP v2, uncheck Use Default

Gateway

VPN login: [keijzer**HYPERLINK "mailto:keijzersm@asml-dmz.local"s**HYPERLINK "mailto:keijzersm@asml-dmz.local"m@asml-dmz.local](mailto:keijzersm@asml-dmz.local) pw: as ASML

JIRA: [https://asml-jira.altran.nl](https://mail.altran.com/owa/redir.aspx?C=m_QaWJ7q5AELsjxNpzh9uGMmmdxtv3CwhvZPomdBXgMAbzSs-fvTCA..&URL=https%3a%2f%2fasml-jira.altran.nl)

SVN: [https://asml-svn.altran.nl/svn/asml](https://mail.altran.com/owa/redir.aspx?C=Y1dW8AHBuQn4HFBLau9dKYkYYTwyyQk61T7PbJySoi8AbzSs-fvTCA..&URL=https%3a%2f%2fasml-svn.altran.nl%2fsvn%2fasml)

Change password: [https://asml-aed.altran.nl](https://mail.altran.com/owa/redir.aspx?C=6ausiqJ14Z77ZgVqUfCiZSL0UTWaC51MzHXXH0S31V0AbzSs-fvTCA..&URL=https%3a%2f%2fasml-aed.altran.nl)

**Contactpersonen**

Rene Kempen FI FC29

Anil Kumar Thimmaiah FC29/Twist

Surya Modadugu FC29/Twist

**ASML General**

Salary number 00314100

Group DE SW/MET Outsourced Activities

**Telephone numbers**

Helpdesk: 040-268 5555

Badge etc 040-268 4444

**Unix**

Create new terminal: xterm &

Tar extract gzip –d file; ?

tar –xvf <file>

Find file find . –name ‘\*FilePart\*’ (do not forget ‘ ‘ )

**Websites**

Red Hat Enterprise Portal <https://sswe-migr.asml.com/ovirt-engine/userportal/#basic>

Citrix XenApp Desktops <http://146.106.236.198/Citrix/CitrixPortal/site/default.aspx?CTX_CurrentTab=Desktops>

ASML Portal <https://portal11.asml.com/>

RSA Like ING

ASML VPN <http://146.106.236.198/Citrix/CitrixPortal/auth>

ASML VPN (alt) <http://146.106.236.198/Citrix/CitrixPortal/auth/login.aspxJan>

VDI [https://sswe.asml.com/UserPortal/org.ovirt.engine.ui.userportal.UserPortal/UserPortal.html#login](https://sswe.asml.com/UserPortal/org.ovirt.engine.ui.userportal.UserPortal/UserPortal.html)

Code Collaborator <https://reviewtool-eu.asml.com:9020/ui>

TCE <http://tc-eu.asml.com/tc/webclient> , use group: , rol:

Jira <https://jira.tassprojects.nl:8443/secure/Dashboard.jspa>

Jira (alt) https://jira-asml.tassprojects.nl

KT Devbench [http://techwiki.asml.com/index.php/KT\_\_Logical\_Action\_Layer\_Simulator/FAQ#How\_to\_handle\_wafers](http://techwiki.asml.com/index.php/KT__Logical_Action_Layer_Simulator/FAQ)

Test Arena <http://testarena.asml.com:8080/TestArena>

Holiday plan <http://holiday.asml.com/cgi-bin/hday_editor.pl>

Fishbone [https://realtimeboard.com](https://realtimeboard.com/)

**Clearcase**

Create dumbo: ccmk\_dumbo –r at6.1\_pc6404 test

Create view: ccjoin\_proj (-I = integrator mode) –l (latest baseline) (-b baseline) project

e.g. ccjoin\_proj at6.1\_pc6404

Buildscope ccget\_buildscope, ccset\_buildscope

Scope sort ccrscope -p

Tree xclearcase –vtree <file>

Patch info: ccget\_config

Workbench wrwb & (-r | -R)

Workbench getting toolbar Select another perspective and adapt

Activities clearquest

Changed files lsco

Password change: crmregister add –database SWCHG –connection CQMS.ASML.EINDHOVEN –user

mickeijz –password …

rmname Indien niet lukt (file uitgechecked, maar in onbekende stream): gebruik rmname –f

Om te checken bij wie: cleartool lsco name

**ClearQuest**

Find change Menu option in Edit menu

**View**

Inspect code (diffs) ccinspect

**Patches**

Make devpatch ccmk\_devpatch

**FI**

Unlock stream ccchg\_lock unlock <user>

User names ccadmin cyiek emichiel estens everhuls mickeijz mvannieu simgt simgt\_tl smodadug yuguan

Change FI: ccchg\_fi -n mickeijz .

Check baseline

Make new baseline int str: 6.1: /home/mvannieu/public/bin/FI –b && ccupd\_modcomps

/recommend bl 6.2: ccmk\_bl && ccset\_bl\_stat –P BUILT && ccset\_bl\_stat –r && ccupd\_modcomps

Ccmake/Update scope /home/mvannieu/public/bin/FI -u

Int stream interfaces: ccupdate\_xidl; ccupdate\_scope -a

Remove exports: ls –lrt /exp/at6.2\_pb5557

ccrm\_export –f <name>

Check feature integrator /home/jleeuwes/bin/ccget\_fi

**Sync**

Sync check: /home/mvannieu/public/bin/project\_numbers.sh

Sync between release ccmerge –m (-ch) –a <delivery\_patch> (SWCHG00847453 for QBL)

Delivery patch: Select any file changed that is delivered, go to destination path in vtree, select looking glass,

Check ‘Attached Activity’ (2/3 of text).

**View**

Show errors cclst\_builderr

Info /home/jkrielen/bin/cqinfo <CC/BB), use –b for BB or –h for help

**IDE**

Start: Internet explorer, VDI link, make console, type ‘wrwb &’

Link problems ln –s <file> in tst file

**TCE**

Make Ucc Check group lead first

Change group lead Make sure UCC request not in progress, select green book icon document, right click right on the arrow on the document, select properties, select All, Press Check-Out and Edit, change ASML\_GroupLeader item, click Check-In/Save.

**SIA**

**TPS**

Doxygen generation: /home/dheesch/lnx/bin/run\_doxygen –t –o ~/<output filder> <CC>

Files are in h:\Download\Dox…\doxygen … …doxygen doxfile

**TAR**

ECCE Test Create DB, non KT (devbench create –M NXT:1980Di DB\_ECCE) from view, and activate

emcf\_recode CM -I INFORMPRO2 -v ENABLED -o . -r .

emcf\_recode CM -i EDA\_INTERFACE -v ENABLED -o . -r .

Start twinscan and initialize drivers

Ifconfig –a to give IP address

Let FC29 (e.g. Ad Smits) perform ECCE test, send IP address

**Alternative: Run EI regression test**

Mail Ad Smits or Chuen Yiek

Valgrind Create KT devbench,: devbench lnxcreate -t KT -p -M NXT:1950Ai DevbenchName

Activate + dhh\_ready

Sync test files: devbench sync –t –p –l DevBenchName

Create Valgrind file: echo "--leak-check=full --num-callers=30" > bin/lnx/KDMA.valgrind

From 1950 test arena scenario xml file:

Install metro: /sdev/user/bin/metro\_autotest –i K

Start: KT/tstpkg/KT\_CC\_ensure\_system\_start.sh –c KD

Run tests: /sdev/user/bin/metro\_autotest –r KD

Stop: KT/tstpkg/KT\_system\_stop -sf; /sdev/user/sbin/kill\_atlas;

/usr/asm/atl.0000/bin/sun/install/asm\_upgrade\_const -o SYSTEM\_STOP -s

Check info in valgrind

**TDL** In Codecollab, press Edit to change Link to TDL-tool to correct number (TDL-XXXXXX), go to link, save file, and check in under TDL in TCE.

**Code Collaborator**

Link <https://reviewtool-eu.asml.com:9020/ui>

Code review /home/mvannieu/public/bin/ccollab\_upload

**IDE**

Tics / toolbar / analyze

**DevBench**

Create KT devbench devbench create –t KT –p –M NXT:1970Ci <devbench\_name>

or via test arena, fc28 clean\_environment scenario (use baseline as source)

Create non KT devbench create <devbench\_name>

Activate devbench active, in terminal: tcsh

devbench lnxactivate -X

In activate, go down, select left bottom icon (start menu), select System Tools,

LXTerminal

Prepare KB emcf\_recode CM -i INFORMPRO2 -v ENABLED -o . -r .

emcf\_recode CM -i INTRAFIELD\_WAFER\_ALIGNMENT -v EXTENDED -o . -r .

check: emcf\_view

cp /home/mickeijz/Public/InformPro2/KT\_wafers\_in KT

cp /home/mickeijz/Public/InformPro2/OV\_IFWA\_AR\* user\_data/RDM/

Start twinscan start, after driver init: Configuration:

Factory Constant: KD Wafer Alignment: Grid residual outlier detection :False -> True

Run Lot Select lot, select reticle, second layer, 1 wafer.

Stop KT Twinscans KT/tstpkg/KT\_CC\_ensure\_system\_stop.sh

Run test KD/tstpkg/KDAL\_test\_model\_data

Purify Instrument: /home/apashev/bin/ccpurify.lnx –t mickeijz\_DB3 KDMA

Copy 2 DB: devbench dump mickeijz\_DB3 bld/KDMA bin/sun/KDMA

In DB: start

View: /home/apashev/bin/ccpurify.lnx –view <pvfile> (copy to home dir)

Code complexity: (McCabe): /home/mfrijns/bin/pmccabe\_linux <filename>

Trace output: cat /usr/asm/data.0000/output\_informal\_diagnostics/TH/trace/ | grep “{MK” >out…

Switch on trace:

Result in trace “%R”,

Even log diff: cd /home/bspoor/quick-analysis-tool/bin/

ER\_event\_log\_tool.py list\_unique --help

MDL cd (to go to base dir)

MDL\_view /data/diagnostics/ME/… met current lot > output.txt

Tags: KD-C115: Alignment Smash periodic recipe info

KD-C102: field positions IFWA marks

KD-C90E: Smash periodic color scan results

KD-C90D:

KA-C902: Reject list IFWA

KA-CA09: Reject list FIWA

Shared drive [\\asml.com\eu\shared\nl011012\_u\altwin\SMASH\_Mk-X\Users\Jorn\_Lucas-JLUE](file://\\asml.com\eu\shared\nl011012_u\altwin\SMASH_Mk-X\Users\Jorn_Lucas-JLUE)

THCC\_set\_mode –o JLUE –r test AS 0 0 0

Python tb\_cadenv\_exec python /home/atlevel/Focus\_Application/recipes\_and\_queues/TWINSCAN/devbench\_scripts/TI\_set\_reticle.pyc -r 45564391A001 -b 1 -s 5 -i

ADEL On Firefox (EID): <http://devbench-litho-eid-prd.sn-eu.asml.com:8080/devbench/>

Select devbench, menu ‘ More’, EDI, than add ADELalignmentReport/ (don’t forget /)

Or <http://172.31.53.228:8080/EDI/ADELalignmentReport/>

Get internet address by ifconfig –a

Unencrypting ADEL: <http://techwiki.asml.com/index.php/YLKD>

AID: <http://172.31.53.228:8080/AID> (or aid?), tooling, 1qazxsw2

TPT: grep /usr/asm/data.0000/output\_informal\_diagnostics/TH/data | grep “KD-3000”

Twist test cases manually:

KD test cases manually: Change in KD\_autotest.xml the set of tests (last part of file)

Follow KD\_CC\_KD\_autotest\_NXT\_2950Ai:

* /sdev/user/bin/metro\_autotest -i KD
* KT/tstpkg/KT\_CC\_ensure\_system\_start.sh -c KD

Debugging go to dir to instrument (i.e. ccd KD com int bin

Instrument: ccmake clean && ccmake raw\_debug

Sync devbench lnxsync -c c -l -t mickeijz\_DB5 (-l = l from Lima)

Sync sources /home/agestel/bin/dbsslnx mickeijz\_DB32

In devbench: tb\_cadenv\_exec

Check ls sources

Start start

Run unittest KD/tstpkg/KDtest\_model\_data\_parameters

Stop for debugging int I; while I {}

Check process ps -ef | grep “KDMA” …or model…

Gdb KDMA gdb bin/lnx/KDMA 19945

Gdb unittest gdb KT/tstpkg/KDtest… <nr>

Set var set var i = 0; n

Set sources dir sources

Continue c

See <http://darkdust.net/files/GDB%20Cheat%20Sheet.pdf>

**IBI**

FCO Generation Tool FGT: <http://techwiki.asml.com/index.php/FGT>

Create DB

Install needed pre-patches

**asm\_upgrade\_config -o mickeijz -r for\_testing\_ibi (or fgt)**

**asm\_upgrade\_const -o mickeijz**

**create file install\_input.xml**

* + <install\_input>
    - <install\_patch>AT\_5.0.0.c\_5860a</install\_patch>
  + </install\_input>

See (QBL: /exp/s2p/patch/<user\_name>/\* . 6.1: /exp/at6.1/Patches

**python /log/simgt/TWINRUNR/bin/twinrunr.py --suite SYSTEM --set FGT**

Check wiki for results/copying

IBI <http://techwiki.asml.com/index.php/Ibi>

As above, python command:

**python /log/simgt/TWINRUNR/bin/twinrunr.py --suite SYSTEM --set IBI**

Result: [\\asml.com\eu\shared\nl011022\_u\test\_data\TWINRUNR\IBI\mGF48\mickeijz\_IBI3\_20170705\_171415\result.html](file:///\\asml.com\eu\shared\nl011022_u\test_data\TWINRUNR\IBI\mGF48\mickeijz_IBI3_20170705_171415\result.html)

FCO ME PP <http://collaboration.asml.com/teams/cs-competency-engineering/metrology/_layouts/PowerPoint.aspx?PowerPointView=ReadingView&PresentationId=/teams/cs-competency-engineering/metrology/Shared%20Documents/ME_EE_%20matrix_for_%20document_%20review%20%28FC-061%20and%20FC-062%29.pptx&Source=http%3a//collaboration.asml.com/teams/cs-competency-engineering/metrology/Shared%2520Documents/Forms/AllItems.aspx&DefaultItemOpen=1&DefaultItemOpen=1>

**TestBench/Proto**

Log in remote /log/simgtproto /tooling/proto TB39

Reboot check\_machine\_status | more : shows release (for disk / patch checking)

6.1: reboot (not in su), 6.2: asml\_reboot (not in su)

<http://techwiki.asml.com/index.php/How_To_Change_Disks_On_A_Test_Bench>

Start setMachineType DEFAULT ; TI\_system\_start; TI\_system\_stop

list patches install\_patch list

Install regular patch Sola install\_patch (-f) /exp/at6.2/Patches/AT\_6.2.0.b\_<patchnummer>-

<timestamp>.tgz

Install dev patch QBL: Install\_patch <name\_of\_person>/<name of patch> without tgz

??: install\_patch (-f) /exp/s2p/patch/<username>/<patchname>.tgz

Linux Alpha Patch install asml\_patch install AT\_6.2.0.a\_9039a (using su –)

Asml\_patch install /exp/at6.2/Patches/AT\_6.2.0.b\_5729a-20170619\_161921.tgz

Alpha patch gen\_patch\_on\_SP –r at5.0.0 –p 5860

Alternative Linux/non QBL su - (do not forget -)

asml\_install patch ... (as above).

asml\_patch install ...

Backout patch install\_patch backout

Init return from su (exit): SetMachineType DEFAULT

**Patch Generation Page**

Metroart metro\_test\_submit (/home/metro\_test/tools/metro\_test\_submit) 95262

metro\_test\_advice xxxx (/home/metro\_test/tools/metro\_test\_advice )

Dat+ (dat Plus) ccrun\_datplus

Dat+ On 5.0: cadenv python

setenv PATH /home/mickeijz/.caddir/RHEL6/cadbin:$PATH

python –V (should be 2.7)

ln -s /home/mickeijz/.caddir/RHEL6/cadbin/python /home/mickeijz/.caddir/RHEL6/cadbin/python2.7

**Patch Info FC-61**

EXTIF: D000402025-07-GID-001

SIA: D000396861-01-SIA-001

TPS: D000450148-00-TPS-001

TPS Jorn: D000400876

TAR 6.1 D000486362

SWCHG: SWCHG00681040

TAR 6.2 D000506427

Deliver activity to 6.2 SP10: SWCHG00847453

QBL

Change:

WI: SWCHG00717567

Stream at6.1\_pc6404

Code review: 176228

PGP <http://seitooling.asml.com/cgi-bin/patch_generate/patch_generate.pl?patch=26745>

**6.2 BBs**

SWChange SWCHG00758757

Patch 5557

**QBL**

SWChange SWCHG00848325

Stream at\_qbl\_sync\_620\_pb5557

TAR D000548049/01

Air issue: SX smoke: P310916.

**LO Changes (Ext Ifwa)**

6.1 at6.1\_pc6673

TAR: D000532484

6.2 Code reviews:

at6.2\_pb5729\_int (hjochems)

TAR: D000548045

QBL Code review: 193477

Mickeijz\_at\_qbl\_sync\_620\_pb5729

TAR: D000548049, review: <https://reviewtool-eu.asml.com:9020/ui#review:id=195372>

SWCHG:

**5.0**

Integrator GHEX Gert Heijenk

**BB Name Approved by**

BB-029-0003A EDI gcamps

BB-029-0013A Production Control Data Interfaces fwijnen

BB-040-0001A scm swaard

BB-040-0003A System tests aveldman

BB-040-0012A Configuration 300 NXT Wet dkleef

BB-061-0001A Metro Alnment tcalloar

**Component owners**

SX Coen van Bakel

KD\_LO\_start\_lot

kd\_lo\_start\_lot\_hsa

KD\_PSA\_start\_lot

KD\_PSM\_start\_lot

KD\_rq\_start\_wafer

KD\_LO\_start\_wafer

KD\_LO\_start\_strategy

KD\_CSA\_set\_hsa\_model\_data

KD\_rq\_model\_CSA

KD\_CSA\_model

kd\_csa\_get\_sa\_model\_result

**KD\_LO\_LOT:**

if(r == OK)

{

KDAL\_create\_model\_data(&kd\_model\_data);

KD\_FIWA\_set\_model\_data(kd\_model\_data);

if (set\_active\_strategy)

{

/\* Only active strategy must be logged. \*/

KD\_CSA\_set\_hsa\_model\_data(&(kd\_model\_data->adel\_data.model\_data.StageAlignModelData));

}

else

{

/\* TODO: Handle other strategies \*/

}

}

**CSA:**

if (( r == OK ) && modeling\_successful )

{

ADELmeAlignModelData\_SaModelResult\* sa\_result\_data = NULL;

\*kd\_sa\_model\_data\_exists = FALSE;

kd\_csa\_get\_sa\_model\_result(&sa\_result\_data, single\_TIS);

if (sa\_result\_data != NULL)

{

/\* Set the model parameters \*/

KDAL\_model\_data\_set\_CSA\_model\_parameters(sa\_result\_data, TIS1\_TRM, TIS2\_TRM, single\_TIS);

\*kd\_sa\_model\_data\_exists = TRUE;

}

}

**New functions KD\_CSA\_set\_sa\_model\_data**

**Kd\_csa\_get\_sa\_model\_result**

09/20/2017 18:38:51.5629 Machine:LB99 (Rel:6.2.0.b, KDMA [2204], DDXAxVARARRAY.c, ?.?, 577)

SYSTEM ERROR: DD-0001 DEFAULT

Function of component DD called with an incorrect parameter

DDXAxVARARRAY\_set\_length: illegal NULL pointer value for 'varray' parameter.

**09/20/2017 18:38:51.5630 Machine:LB99 (Rel:6.2.0.b, KDMA [2204], KDAL\_ADEL\_model\_data.c, ?.?, 714)**

SYSTEM EVENT: ER-0FFF DEFAULT (linked to DD-0001)

DEACTIVATE: DD-0001

DEACTIVATE: DD-0001

if (hsa\_model\_data != NULL)

{

// Create array element

r = DDXAxVARARRAY\_set\_length(

ADELmeAlignModelData\_SAMODELRESULT\_STR,

hsa\_model\_data->ModelList.array,

hsa\_model\_data->ModelList.len + 1);

if (r == OK)

{

\*sa\_result = &(hsa\_model\_data->ModelList.array[hsa\_model\_data->ModelList.len]);

}

}

**Ln 714: ERXA\_COND\_DEACT(OK, r, ("KDAL\_create\_SA\_model\_result failed"));**

**ERROR**

r = DDXAxVARARRAY\_set\_length(

ADELmeAlignModelData\_SAMODELRESULT\_STR,

&(hsa\_model\_data->ModelList),

hsa\_model\_data->ModelList.len + 1);

ADELmeAlignModelData\_SaModelData\* hsa\_model\_data

DDF:

typedef ADELmeAlignModelData:SaModelResult ADELmeAlignModelData:SaModelResults [\*];

#--- SA model data ---

typedef struct

{

ADELmeAlignModelData:SaModelResults ModelList

<microhelp = "Stage alignment attempt list data.">;

} ADELmeAlignModelData:SaModelData

**OK**

// Set length to 2 (X, Y).

r = DDXAxVARARRAY\_set\_length(

ADELmeAlignModelData\_FIWAMARKACCEPTANCESUMMARYLISTTYPE\_STR,

summary\_list,

2);

ADELmeAlignModelData\_FiwaMarkAcceptanceSummaryListType\* summary\_list =

typedef ADELmeAlignModelData:FiwaMarkAcceptanceSummaryType ADELmeAlignModelData:FiwaMarkAcceptanceSummaryListType[\*];

static ASML\_result kdal\_process\_modeled\_parameters();

/\* #MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data); \*/

static ASML\_result kdal\_process\_fiwa\_modeled\_parameters();

/\* #MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data); \*/

static ASML\_result kdal\_process\_ifwa\_modeled\_parameters();

/\* #MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data); \*/

static ASML\_result kdal\_compute\_and\_set\_polynomials(

/\* #MK ADELmeAlignModelData\_ModelDataType\* const adel\_model\_data\_p, \*/

const kdal\_coefficient\_struct fiwa\_coefficients[]);

static ASML\_result kdal\_proces\_mark\_acceptance\_list(

/\* #MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data, \*/

kdal\_ifwa\_groups\* const ifwa\_groups);

static ASML\_result kdal\_process\_mark\_acceptance\_list\_for\_fiwa(

/\* #MK KDAL\_KD\_model\_data\* const model\_data, \*/

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data);

static ASML\_result kdal\_process\_mark\_acceptance\_list\_for\_ifwa(

/\* #MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data, \*/

kdal\_ifwa\_groups\* const ifwa\_groups);

static void kdal\_calculate\_mark\_acceptance\_lists(

/\*MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data, \*/

kdal\_ifwa\_groups\* const ifwa\_groups);

void KDAL\_create\_fiwa\_model\_data(

/\* #MK KDAL\_KD\_FIWA\_model\_data\*\* model\_data); \*/

void KDAL\_destroy\_fiwa\_model\_data(

/\* #MK KDAL\_KD\_FIWA\_model\_data\*\* model\_data); \*/

I void KDAL\_create\_model\_data(

/\* #MK KDAL\_KD\_model\_data\*\* model\_data); \*/

void KDAL\_destroy\_model\_data(

/\* #MK KDAL\_KD\_model\_data\*\* model\_data); \*/

void KDAL\_model\_data\_set\_strategy\_id(

/\* #MK KDAL\_KD\_model\_data\* model\_data, \*/

DTXA\_align\_strategy\_id\_string strategy\_id);

void KDAL\_model\_data\_store\_fiwa\_model\_data(

/\* #MK KDAL\_KD\_model\_data\* model\_data,

KDAL\_KD\_FIWA\_model\_data\* fiwa\_model\_data); \*/);

void KDAL\_create\_SA\_model\_result(

/\* #MK Check \*/

ADELmeAlignModelData\_SaModelData\* hsa\_model\_data,

ADELmeAlignModelData\_SaModelResult\*\* sa\_result);

// External Parameters functions

void KDAL\_model\_data\_set\_fiwa\_parameters(

/\* #MK KDAL\_KD\_FIWA\_model\_data\* model\_data, \*/

const KDWAXAxPOLYNOM\_polynome \* const fiwa\_polynome);

void KDAL\_model\_set\_FIWA\_results(

/\* #MK KDAL\_KD\_FIWA\_model\_data\* model\_data, \*/

ASML\_bool valid);

void KDAL\_model\_data\_set\_ifwa\_parameters(

/\* #MK KDAL\_KD\_FIWA\_model\_data\* model\_data, \*/

const KDWAXAxGRIDxWG\_field\_grid \* const ifwa\_field\_grid);

// Mark Acceptance and Mark Reject functions

void KDAL\_model\_data\_store\_fiwa\_mark(

/\* #MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data, \*/

int mark\_id,

ASML\_bool is\_x\_scan,

ASML\_bool is\_y\_scan);

void KDAL\_model\_data\_store\_ifwa\_mark(

/\* #MK KDAL\_KD\_model\_data\* const model\_data,

KDAL\_KD\_FIWA\_model\_data\* const fiwa\_model\_data, \*/

int mark\_id,

ASML\_bool is\_x\_scan,

ASML\_bool is\_y\_scan,

xyvect field\_grid\_pos);

void KDAL\_write\_instance(

/\* #Mk KDAL\_KD\_model\_data\* model\_data); \*/);

**kd\_fiwa\_adel\_writing\_enabled**

**In KD\_LO\_start\_wafer:**

KD\_FIWA\_set\_adel\_writing\_enabled(kd\_lo\_is\_adel\_writing\_enabled());

**static ASML\_bool kd\_lo\_is\_adel\_writing\_enabled(void)**

{

ASML\_bool is\_adel\_writing\_enabled = (kd\_lo\_lot\_data.active\_strategy.marks.len > 0);

int index = 0;

for (index = 0; (index < kd\_lo\_lot\_data.nr\_evaluation\_strategies) && (!is\_adel\_writing\_enabled); index++)

{

is\_adel\_writing\_enabled = (kd\_lo\_lot\_data.evaluation\_strategy[index].marks.len > 0);

}

return is\_adel\_writing\_enabled;

}

**void KD\_FIWA\_set\_adel\_writing\_enabled(ASML\_bool adel\_writing\_enabled)**

{

kd\_fiwa\_adel\_writing\_enabled = adel\_writing\_enabled;

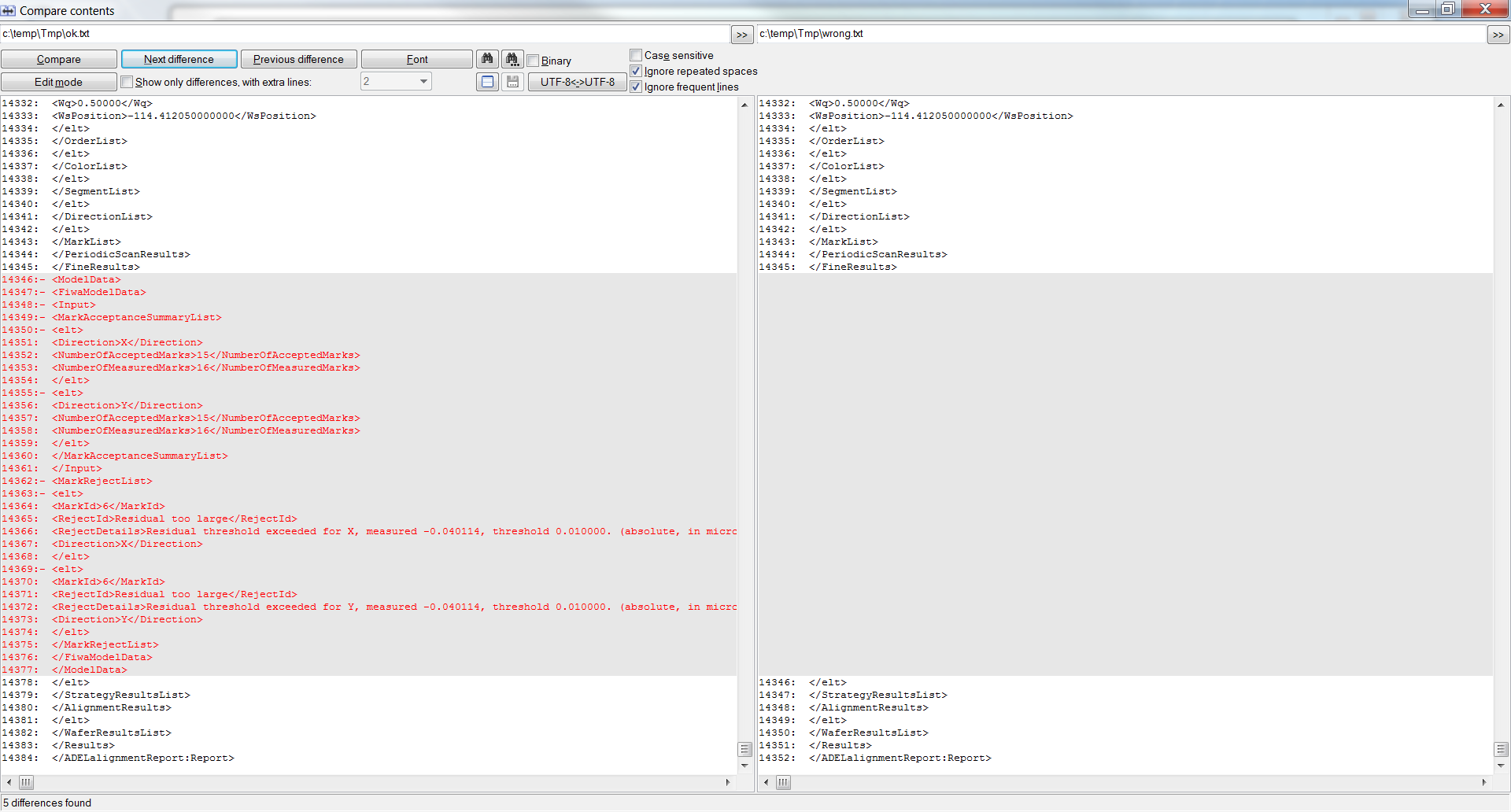
if (kd\_fiwa\_model\_data != NULL)

{

KDAL\_destroy\_model\_data(&kd\_fiwa\_model\_data);

}

}



*<configuration\_id>NXT\_1980Di\_KT Configuration\_1</configuration\_id>*

*<environment>DEVBENCH</environment>*

*<machine\_type>NXT:1980Di</machine\_type>*

*<test\_package>ASML-BB-029-0013A-Test</test\_package>*

*<test\_case\_set>*

*<pre\_test\_case\_set\_script>*

***KT/tstpkg/KT\_install.sh -m NXT:1980Di****</pre\_test\_case\_set\_script>*

*<pre\_test\_case\_set\_script>*

***ksh* /log/simgt/TWINRUNR/TstSuites/TstScripts/SYSreliab/dat/SX\_machine\_const -linkedlitho**

</pre\_test\_case\_set\_script>

<pre\_test\_case\_set\_script>

**/usr/asm/atl.0000/YLKD/tstpkg/TstWare/TstData/script/disable\_encryption.sh**

</pre\_test\_case\_set\_script>

<test\_case>

<test\_case\_id>InformPro2 - FC-061 - IFWA\_ADELalignmentReport\_3a</test\_case\_id>

<max\_duration>1800</max\_duration>

<pre\_exec\_script>**emcf\_recode CM -i INFORMPRO2 -v ENABLED -o . -r .**

</pre\_exec\_script>

<pre\_exec\_script>**emcf\_recode CM -i INTRAFIELD\_WAFER\_ALIGNMENT -v ENABLED -o . -r .**

</pre\_exec\_script>

<pre\_exec\_script>KT/tstpkg/KT\_start.sh;ls -la KT</pre\_exec\_script>

<pre\_exec\_script>

**yes | cp /usr/asm/atl.0000/YLKD/tstpkg/TstWare/TstData/KT\_wafers\_in/KT\_wafers\_in\_test1 /usr/asm/atl.0000/KT/KT\_wafers\_in**

</pre\_exec\_script>

<pre\_exec\_script>**KT\_read\_data**</pre\_exec\_script>

<pre\_exec\_script>

**FCclip\_apply -d KD/KD\_factory.const -c /usr/asm/atl.0000/YLKD/tstpkg/TstWare/TstData/clipfiles/IFWA\_ADELalignmentReport.clip -o TWIST -r TWIST**

</pre\_exec\_script>

<exec\_script>

**/shared/nl011012/twist/release/v2.3.46/bin/tb\_twist\_exec --confdir /usr/asm/atl.0000/YLKD/tstpkg/TstWare --conffile TestEnvironmentConfig.xml --target twinscan --track TT\_IFWA\_ADELalignmentReport\_3 --vector TV\_IFWA\_ADELalignmentReport\_3a --continueonerror**

</exec\_script>

<post\_exec\_script>TPF\_SYSTEM\_STOP</post\_exec\_script>

<on\_error\_script>TPF\_SYSTEM\_STOP</on\_error\_script>

<on\_error\_timeout\_script>TPF\_SYSTEM\_STOP</on\_error\_timeout\_script>

</test\_case>

<test\_case>

<test\_case\_id>InformPro2 - FC-061 - IFWA\_ADELalignmentReport\_5</test\_case\_id>

<max\_duration>1800</max\_duration>

<pre\_exec\_script>KT/tstpkg/KT\_start.sh;ls -la KT</pre\_exec\_script>

<exec\_script>/shared/nl011012/twist/release/v2.3.46/bin/tb\_twist\_exec --confdir /usr/asm/atl.0000/YLKD/tstpkg/TstWare --conffile TestEnvironmentConfig.xml --target twinscan --track TT\_IFWA\_ADELalignmentReport\_5 --vector TV\_IFWA\_ADELalignmentReport\_5 --continueonerror</exec\_script>

<post\_exec\_script>TPF\_SYSTEM\_STOP</post\_exec\_script>

<on\_error\_script>TPF\_SYSTEM\_STOP</on\_error\_script>

<on\_error\_timeout\_script>TPF\_SYSTEM\_STOP</on\_error\_timeout\_script>

</test\_case>

// Empty instances used to prevent performing a full create more than once.

static KDAL\_KD\_adel\_model\_data\* kdal\_empty\_adel\_model\_data = NULL;

static KDAL\_KD\_fiwa\_ifwa\_model\_data kdal\_empty\_fiwa\_ifwa\_model\_data = NULL;

// Working instance used for filling/sending to DP.

static KDAL\_KD\_adel\_model\_data\* kdal\_working\_adel\_model\_data = NULL;

// Working instance used for temporary FIWA/IFWA storage.

static KDAL\_KD\_fiwa\_ifwa\_model\_data kdal\_working\_fiwa\_ifwa\_model\_data = NULL;

Debugging go to dir to instrument (i.e. ccd KD com int bin

Instrument: ccmake clean && ccmake raw\_debug

Sync devbench lnxsync -c c -l -t mickeijz\_DB5 (-l = l from Lima)

Sync sources /home/agestel/bin/dbsslnx mickeijz\_DB32

In devbench: tb\_cadenv\_exec

Check ls sources

Start start

Run unittest KD/tstpkg/KDtest\_model\_data\_parameters

Stop for debugging int I; while I {}

Check process ps -ef | grep “KDMA” …or model…

Gdb KDMA gdb bin/lnx/KDMA 19945

Gdb unittest gdb KT/tstpkg/KDtest… <nr>

Set var set var i = 0; n

Set sources dir sources

Continue c

**title Wafer Alignment**

**loop Each Lot start**

**alt ADEL writing enabled**

**KD\_LO\_lot.c->KD\_LO\_lot.c:kd\_lo\_process\_wafer\_alignment\_settings\_data()**

**alt statistics filled**

**KD\_LO\_lot.c->KDAL\_ADEL\_wafer\_alignment: KDAL\_ADEL\_alignment\_settings\_create(adel\_alignment\_settings, nr\_of\_strategies)**

**KDAL\_ADEL\_wafer\_alignment->DDXA: DDXAxOBJECT\_full\_create(alignment\_settings);**

**KDAL\_ADEL\_wafer\_alignment->DDXA: DDXAxOBJECT\_set\_legnth(alignment\_settings, nr\_of\_strategies);**

**KD\_LO\_lot.c->KDAL\_ADEL\_wafer\_alignment: KDAL\_ADEL\_store\_lot\_data(adel\_alignment\_settings, lot\_data, lot\_info);**

**loop strategies**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: store\_lot\_data\_for\_strategy(lot\_data, lot\_info, adel\_alignment\_settings evaluation strategy)**

**end**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: store\_lot\_data\_for\_strategy(lot\_data, lot\_info, adel\_alignment\_settings active strategy)**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_store\_verified\_mark\_usage\_data(marks, strategy\_settings);**

**for mark**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_store\_mark\_usage\_data\_for\_coarse\_and\_fine(mark\_index, mark\_spec, strategy\_list);**

**alt coarse mark**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_store\_mark\_usage\_list\_data\_for\_coarse\_settings(mark\_id, mark\_spec, coarse\_mark\_list;**

**else fine mark**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_store\_mark\_usage\_list\_data\_for\_fine\_settings(mark\_id, mark\_spec, fine\_mark\_list;**

**end**

**end**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_set\_fiwa\_settings(current\_strategy, strategy\_settings);**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_calculate\_number\_of\_measurements\_fiwa(source\_marks, nr\_of\_fiwa);**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_store\_threshold\_values(strategy\_settings, nr\_fiwa, current\_strategy);**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_set\_ifwa\_settings(marks, lot\_info, current\_sttrategy, ifwa\_settings);**

**KDAL\_ADEL\_wafer\_alignment->DDXA: DDXAxOBJECT\_sub\_create(marks);**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_fill\_marks(current\_strategy, marks);**

**KDAL\_ADEL\_wafer\_alignment->DDXA: DDXAxOBJECT\_sub\_create(ext\_ifwa\_settings);**

**KDAL\_ADEL\_wafer\_alignment->KDXAxIFWAxLIB\_: KDXAxIFWAxLIB\_determine\_ext\_ifwa\_parameters(.., marks, ext\_ifwa\_settings);**

**KDAL\_ADEL\_wafer\_alignment->KDAL\_ADEL\_wafer\_alignment: kdal\_copy\_n\_out\_of\_m\_list(ext\_ifwa\_settings\_n\_out\_of\_m\_list, ifwa\_settings\_n\_out\_of\_m\_list);**

**KDAL\_ADEL\_wafer\_alignment->DDXA: DDXAxOBJECT\_sub\_destroy(marks);**

**KDAL\_ADEL\_wafer\_alignment->DDXA: DDXAxOBJECT\_sub\_destroy(ext\_ifwa\_settings);**

**KD\_LO\_lot.c->KD\_LO\_lot.c: kd\_lo\_ADEL\_store\_ext\_ifwa\_settings**

**loop strategies**

**KD\_LO\_lot.c->KD\_LO\_lot.c: kd\_lo\_ADEL\_store\_ext\_ifwa\_settings\_for\_strategy**

**KD\_LO\_lot.c->KDAL\_ADEL\_wafer\_alignment: KDAL\_ADEL\_store\_intrafield\_extended\_recipe\_settings**

**end**

**KD\_LO\_lot.c->KDAL\_ADEL\_wafer\_alignment: KDAL\_ADEL\_log\_wafer\_alignment\_data(adel\_alignment\_settings)**

**KDAL\_ADEL\_wafer\_alignment->DPxAP: DPxAP\_write\_instance(adel\_alignment\_settings);**

**KD\_LO\_lot.c->KDAL\_ADEL\_wafer\_alignment: KDAL\_ADEL\_alignment\_settings\_destroy(adel\_alignment\_settings)**

**KDAL\_ADEL\_wafer\_alignment->DDXA: DDXAxOBJECT\_full\_destroy(alignment\_settings);**

**end**

**end**

**title ADEL Alignment**

**LO->KD\_SMCOWA\_smash: KD\_SMCOWA\_set\_align\_result(scan\_id, raw\_align\_result);**

**KD\_SMCOWA\_smash->KDAL\_ADEL\_alignment: KDAL\_send\_COWA\_mark\_SMASH(cowa\_mark, cowa\_mark\_id, scan\_type, scan, align\_result\_base, detailed\_raw\_result);**

**alt Smash Periodic, template, multiperiodic**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_mark\_periodic(mark, cowaMark, cowaMarkId, scan, align\_result\_base, periodic\_recipe);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setStrategyId(mark);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_get\_mark\_usage(usage);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_set\_periodic\_scan\_aligned\_position**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_allocate\_var\_array**

**KDAL\_ADEL\_alignment->DDXA: DDXAxVARARRAY\_set\_length(length);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_mark\_periodic\_reject\_info(reject\_info, dir\_x, dir\_array\_x, scan, dir\_type\_x, measured\_position\_x);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_periodic\_direction\_recipe**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_getPeriodicNrEnabledSegments(result\_struct);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_allocate\_var\_array(segment\_list)**

**loop period**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_periodic\_segment(adel\_struct, result\_struct, adel\_reject\_info);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setSegmentId(segment, segment\_id);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_getPeriodicNrEnabledColors**

**loop color**

**kdal\_fill\_periodic\_color(color, color\_index, result\_struct, adel\_reject\_info);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setColorId(color, color\_index);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_getPeriodicNrEnabledOrders(color)**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_allocate\_var\_array**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_periodic\_order(color\_index, color, order, order\_index, adel\_reject\_info);**

**loop order**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setOrderId(order, order\_id);**

**alt adel\_reject\_info (non multi periodic order)**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_adel\_reject\_info(color, os, adel\_reject\_info)**

**alt valid, enabled**

**alt delta shift failed**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id**

**else:**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_adel\_non\_delta\_shift\_reject\_info**

**alt weight factors small enough**

**alt fit failed:**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info. prio\_fit\_failed, FitFailed);**

**alt capture failed:**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info, prio\_capture\_failed, CaptureFailed);**

**alt status\_ok:**

**alt eval\_status rejected by wq thresholding**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info, wafer\_quality\_too\_low, WaferQualityTooLow);**

**alt eval status rejected by MCC thresholding**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info, mcc\_too\_low, MccTooLow);**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_mark\_periodic\_reject\_info(reject\_info, dir\_y, dir\_array\_y, scan, dir\_type\_y, measured\_position\_y);**

**AS ABOVE**

**end**

**LO->KD\_FIWA: KD\_FIWA\_set\_align\_result(scan\_id, raw\_align\_result, fiwa\_ifwa\_combined\_applied)**

**KD\_FIWA->KDAL\_ADEL\_alignment: KDAL\_send\_FIWA\_mark\_SMASH(fiwa\_mark, fiwa\_mark\_id, scan, align\_smash, align\_result\_base);**

**LO->KD\_LO: KD\_LO\_finish\_strategy(...)**

**KD\_LO->KDAL\_ADEL\_alignment: KDAL\_send\_wafer\_parameters(parameters);**

**KDAL\_ADEL\_alignment->KDTR: KDTR\_get\_stage\_parameters(parameters\_e\_ws, parameters\_msx, parameters msy);**

**KDAL\_ADEL\_alignment->KDTR: KDTR\_get\_initial\_wafer\_load\_offset();**

**KDAL\_ADEL\_alignment->KD\_LO: KD\_LO\_get\_wafer\_nr();**

**KDAL\_ADEL\_alignment->DPxAP: DPxAP\_write\_instance(wafer\_parameters);**

**title ADEL Alignment**

**LO->KD\_SMCOWA\_smash: KD\_SMCOWA\_set\_align\_result(scan\_id, raw\_align\_result);**

**KD\_SMCOWA\_smash->KDAL\_ADEL\_alignment: KDAL\_send\_COWA\_mark\_SMASH(cowa\_mark, cowa\_mark\_id, scan\_type, scan, align\_result\_base, detailed\_raw\_result);**

**alt Smash Periodic, template, multiperiodic**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_mark\_periodic(mark, cowaMark, cowaMarkId, scan, align\_result\_base, periodic\_recipe);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setStrategyId(mark);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_get\_mark\_usage(usage);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_set\_periodic\_scan\_aligned\_position**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_allocate\_var\_array**

**KDAL\_ADEL\_alignment->DDXA: DDXAxVARARRAY\_set\_length(length);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_mark\_periodic\_reject\_info(reject\_info, dir\_x, dir\_array\_x, scan, dir\_type\_x, measured\_position\_x);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_periodic\_direction\_recipe**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_getPeriodicNrEnabledSegments(result\_struct);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_allocate\_var\_array(segment\_list)**

**loop period**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_periodic\_segment(adel\_struct, result\_struct, adel\_reject\_info);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setSegmentId(segment, segment\_id);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_getPeriodicNrEnabledColors**

**loop color**

**kdal\_fill\_periodic\_color(color, color\_index, result\_struct, adel\_reject\_info);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setColorId(color, color\_index);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_getPeriodicNrEnabledOrders(color)**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_allocate\_var\_array**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_periodic\_order(color\_index, color, order, order\_index, adel\_reject\_info);**

**loop order**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_setOrderId(order, order\_id);**

**alt adel\_reject\_info (non multi periodic order)**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_adel\_reject\_info(color, os, adel\_reject\_info)**

**alt valid, enabled**

**alt delta shift failed**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id**

**else:**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_adel\_non\_delta\_shift\_reject\_info**

**alt weight factors small enough**

**alt fit failed:**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info. prio\_fit\_failed, FitFailed);**

**alt capture failed:**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info, prio\_capture\_failed, CaptureFailed);**

**alt status\_ok:**

**alt eval\_status rejected by wq thresholding**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info, wafer\_quality\_too\_low, WaferQualityTooLow);**

**alt eval status rejected by MCC thresholding**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_update\_highest\_priority\_reject\_id(reject\_info, mcc\_too\_low, MccTooLow);**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**end**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_mark\_periodic\_reject\_info(reject\_info, dir\_y, dir\_array\_y, scan, dir\_type\_y, measured\_position\_y);**

**AS ABOVE**

**end**

**LO->KD\_FIWA: KD\_FIWA\_set\_align\_result(scan\_id, raw\_align\_result, fiwa\_ifwa\_combined\_applied)**

**KD\_FIWA->KDAL\_ADEL\_alignment: KDAL\_send\_FIWA\_mark\_SMASH(fiwa\_mark, fiwa\_mark\_id, scan, align\_smash, align\_result\_base);**

**KDAL\_ADEL\_alignment->KDAL\_ADEL\_alignment: kdal\_fill\_mark\_periodic(mark, fiwa\_mark, fiwa\_mark\_id, scan, align\_result\_base);**

**AS ABOVE**

**LO->KD\_LO: KD\_LO\_finish\_strategy(...)**

**KD\_LO->KDAL\_ADEL\_alignment: KDAL\_send\_wafer\_parameters(parameters);**

**KDAL\_ADEL\_alignment->KDTR: KDTR\_get\_stage\_parameters(parameters\_e\_ws, parameters\_msx, parameters msy);**

**KDAL\_ADEL\_alignment->KDTR: KDTR\_get\_initial\_wafer\_load\_offset();**

**KDAL\_ADEL\_alignment->KD\_LO: KD\_LO\_get\_wafer\_nr();**

**KDAL\_ADEL\_alignment->DPxAP: DPxAP\_write\_instance(wafer\_parameters);**

**Triggers**

**NoteOn**

trigger <Name> NoteOn <McSpec> <NoteSpec> <ProgramName> <status>

@ means: case insensitive

<Name> = <identifier>

<McSpec> = ( [@“Mc=” | “” ] <Mc>)

<Mc> = ( “1” | “2” .. | “15” | “16” )

<NoteSpec> = ( [@“Notes=” | “”] (@“ALL” | <NoteRange>)) |

( [@“Note=” | “”] <Note>)

<NoteRange> = ( [<Note>] “~” [<Note>] )| where Note1 <= Note2, Note >= “C1”

(first 12 notes are reserved), Note <” G-10”,

“~” means ALL

<Note> = (“C” | “D” | “E” | “F” | “G” | “A” | “B” ) (“#” | “b” | “”) (“-1” | “0” .. “7” | “8”], check value 1<=note <=127

<ProgramName> = <identifier>

<identifier> = <alpha\_char> + <id\_char>\*

<alpha\_char> = (“A”| “B” | ..| “Y” | “Z” | “a” | “b” | .. | “y” | “z” )

<digit> = (“0” | “1” | .. | “8” | “9”

<id\_char> = <alpha\_char> |<digit> | “\_” )

<status> = (@“enabled” | @“disabled”)

trigger NoteOn MC=1~4 Note=C1~G#5 Transpose5 disabled;

**NoteOff**

will be similar

**Cc Trigger**

trigger <name> Cc <McSpec> <CcSpec> <ProgramName> <status>

<BnkPrgCheck> = ( @”BnkPrg” | “” ) ( <Mc> “.” <Bank> “.” <Prg> )

<Bank> = <digit> 0..127

<Prg> = <digit>0..127

**Translation to MC**

Always one MC

If note range = ALL or ~: create trigger in ALL-notes trigger table

Else if note range: create trigger in OCTAVE\_notes trigger table and specific NOTE trigger table \*

Else if single note: create trigger in single NOTE trigger table

\* example Note range F4~G7 results in 7 single-note triggers for F4 to B4 + 2 octave triggers for octave 5 and 6 and 8 single note triggers for C7-G7, total: 7 + 2 + 8 + 17.

The trigger table is defined by a hash key depending on MC, Note. Note can be:

Single note triggers: 0-127

Octave note triggers: 128-137 (octave 1..10)

All note trigger: 255

**When a note is received**

3 hash keys are calculated:

* All notes trigger table (using also MC)
* Octave trigger table (using also MC, note (octave))
* Single note trigger table (using also MC)

For all three tables, all commands are executed (and checked if the trigger is enabled and condition is met since other triggers can end up in the same trigger table).

**Performance**

To loop through 100 triggers within 3 tables, every trigger taking 10 instructions to check, taking 50 clock cycles. These are 50,00 clock cycle. Assuming 168 MHz this will cost 5,000/168,000,000 = 0.029 ms, thus very less.

The execution of the programs (assume 5 on average high), cost 500 instructions of 4 clock cycli each, resulting in 10,000 clock cycli, which results in 0.059 ms (assuming 168 MHz), also vey less.

**Memory usage**

Assuming there are 256 hash keys / trigger tables, and per entry the following information is stored:

* Type (MIDI/PedSw): 3 bits
* Enabled 1 bit
* For e.g. note on: MC 4 bits
* For e.g. note on: Note 8 bits (for oct)
* Program index 16 bits

Total: 32 bits -> 4 bytes

Thus total storage:

* Table start offsets: 256 \* 2 bytes (start) = 512 bytes
* Tables itself: 1,000 (entries) \* 4 bytes = 4,000 bytes

**Send DMX Instructions**

**Code examples**

send DMX Channel 128 Value 10

Send DMX Channel 128 Var V1

Send DMX Channel 128 Prop NoteNumber // Eg from MIDI

send DMX Channel 128 Values 10 20 30 40 50 60 70 80

send DMX Scenes 0 1 2 3 5 7 Mult 16 Value 10

send DMX OffsetChannel 128 Scenes 0 1 2 3 5 7 Mult 16 Value 10

send DMX OffsetChannel 128 Scenes 0 1 2 3 5 7 Mult 16 Values 10 20 30 40 50 60 70 80

**Code instructions**

0: Single channel

“send” “DMX” <channels\_str> <channel> <values\_str> <value>

1: Multiple channels

“send” “DMX” <channels\_str> <channel> <values\_str> (<value>)+

2: Multiple scenes, single value

“send” “DMX” <offset\_channel\_str> <offset\_channel> <scenes\_str> <scene>+ [<mult\_str> (8 | 16)] <values\_str> <value>

3: Multiple scenes, multiple values

“send” “DMX” <offset\_channel\_str> <offset\_channel> <scenes\_str> <scene>+ [<mult\_str> (8 | 16)] <values\_str> <value>+

<channels\_str> = (“channels” | “ch”), case independent

<channel> = <integer 0..255>

<value\_str> = (“values” | “val” | “v”), case independent

<value> = <integer 0..255>

<scenes\_str> = (“scenes” | “sc”), case independent

<scene> = <integer 0..6>

<mult\_str> = (“mult” | “m”), case independent

<mult> = <”8” | “16”>

<offset\_channel> = (“offset\_channel” | “offset\_ch” | “offch” | “och” | “oc”), case independent

<offset\_channel = <integer 0..2040 or 4080>, multiple of <mult>

**Instruction opcodes**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Byte** | **Bits** | **Meaning** | **Values** | **Meaning** | **Description/remarks** |
| 0 | 7~4 | Instruction Type | ???? | Send DMX |  |
| 0 | 3~2 | Sub command | 00 | Single Channel |  |
|  |  |  | 01 | Multiple Channels |  |
|  |  |  | 10 | Multiple scenes, single value |  |
|  |  |  | 11 | Multiple scenes, multiple values |  |
| 0 | 1~0 |  |  |  | Depending on Sub command |

**00: Single Channel**

0: Single channel

“send” “DMX” <channels\_str> <channel> <values\_str> <value>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Byte** | **Bits** | **Meaning** | **Values** | **Meaning** | **Description/remarks** |
| 0 | 1~0 | Value Type | 00 | Value |  |
|  |  |  | 01 | Variable |  |
|  |  |  | 01 | Property |  |
| 1 |  | Channel | 0-255 | Channel |  |
| 2 |  | Value / Property / Variable | 0-255 | Depending on b0.1~0 |  |

**01: Multiple channels**

1: “send” “DMX” <channels\_str> <channel> <values\_str> (<value>)+

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Byte** | **Bits** | **Meaning** | **Values** | **Meaning** | **Description/remarks** |
| 0 | 0~1 | MSB Channel | 00-11 | MSB Channel) | 0, 256, 512 or 1024 |
|  |  |  | 1 | MSB Channel (255-511) |  |
| 1 |  | Channel | 0-255 | Channel |  |
| 2 |  | NrOfValues | 0-255 | Number of values |  |
| 3..NrOfValues |  | Values | 0-255 |  |  |

**10: Multiple channels, single value**

2: “send” “DMX” <offset\_channel\_str> <offset\_channel> <scenes\_str> <scene>+ [<mult\_str> (8 | 16)] <values\_str> <value>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Byte** | **Bits** | **Meaning** | **Values** | **Meaning** | **Description/remarks** |
| 0 | 1~0 | Value Type | 00 | Value |  |
|  |  |  | 01 | Variable |  |
|  |  |  | 01 | Property |  |
| 1 | 7 | Multiplication factor | 0 | 8 |  |
|  |  |  | 1 | 16 |  |
| 1 | 6 | Offset channel or scenes | 0 | Byte 1 are scenes |  |
|  |  |  | 1 | Byte 1 is offset channel |  |
| 1 | 5~0 | If bit 1.6=0: Scenes |  | 6 scenes | (first) 6 scenes |
|  |  | If bit 1.6=1: Offset Channel | 0-255 | Offset channel multiplied by multiplication factor | If mult factor = 8: 0-512  If mult factor = 16: 0-1024 |
| 2 |  | Scenes |  | (LSB) scenes | (last) 8 scenes |
| 3 |  | Value | 0-255 | Value |  |

**11: Multiple channels, multiple values**

3: “send” “DMX” <offset\_channel\_str> <offset\_channel> <scenes\_str> <scene>+ [<mult\_str> (8 | 16)] <values\_str> <value> +

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Byte** | **Bits** | **Meaning** | **Values** | **Meaning** | **Description/remarks** |
| 0 | 1 | Multiplication factor | 0 | 8 |  |
|  |  |  | 1 | 16 |  |
| 0 | 0 | Offset channel or scenes | 0 | Byte 1 are scenes |  |
|  |  |  | 1 | Byte 1 is offset channel |  |
| 1 |  | If 0.0=0: Scenes |  | 8 scenes | (first) 8 scenes |
|  |  | If 0.0=1: Offset Channel | 0-255 | Offset channel multiplied by multiplication factor | If mult factor = 8: 0-2048  If mult factor = 16: 0-4096 |
| 2 |  | Scenes |  | (LSB) scenes | (last) 8 scenes |
| 3 |  | Number of Values | 0-255 | Number of values |  |
| 4..x |  | Values | 0-255 | Values |  |

* + N

**Hardware (Simple)**

Components

* STM32F103C8T6, 512 KB Flash, 20 KB SRAM
* SD Card
* 2x NRF24L01+

Inputs/outputs

* USB (no MIDI)
* SD slot
* 2 x MIDI In
* 1 x MIDI Out
* DMX Out
  + 2 x green: MIDI In
  + 1 x red: MIDI Out
  + 1 x red: DMX Out
  + 1 x blue: power

**Hardware (Max)**

Components

* STM32F446 (?), 1 GB Flash, 192 KB SRAM
* SD card (reader)
* CAN or RS485
* 2x NRF24L01+

Inputs/Outputs

* CAN/RS485
* USB (for MIDI/updates)
* SD slot
* 4x MIDI In
* 2x MIDI Out
* 2x DMX Out
* LEDS:
  + 4 x green: MIDI In
  + 2 x red: MIDI Out
  + 2 x red: DMX Out
  + 1 x blue: power
  + 6 x yellow/red: RF/comm
  + 1 x yellow/red: CAN/RS485

**Setup**

Kronos MIDI OUT -> Mestra MIDI IN 1

Studiologic MIDI OUT -> Mestra MIDI IN 2

Behringer MIDI OUT -> Mestra MIDI IN 3

(KeyTar -> Mestra MIDI IN 4)

Mestra MIDI OUT 1 -> Kronos MIDI IN

Mestra DMX OUT 5 -> DMX Chain

OR

Kronos MIDI OUT -> Mestra MIDI IN 1

Studiologic MIDI OUT -> Behringer MIDI IN

Behringer MIDI OUT -> Mestra MIDI IN 2 (merged with Studiologic)

Mestra MIDI OUT 1 -> Kronos MIDI IN

Mestra DMX OUT 3 -> DMX Chain

Memory for DMX:

128 channels -> 128 bytes

**Messages for DMX:**

Type

**Buffer MIDI**

**PARIS VID**

**6.2:**

**6.3 extif:** D000512016 – 6.3 EXTIF

Anita Verwegen

name = "intern.KD.wafer\_model\_data",

type\_name = "KDxDATAxMARK:MODEL\_DATA",

data\_category = "output\_application\_reports"

ADELov:StrategyIDType strategyId;

# New DP source for IFP2, all Result related parameter will be part of this object

optional ADELmeAlignModelData:ModelDataType model\_data;

typedef struct

{

optional ADELmeAlignModelData:FiwaModelDataType FiwaModelData

optional ADELmeAlignModelData:IfwaModelDataType IfwaModelData

**optional ADELmeAlignModelData:SaModelData** StageAlignModelData

} ADELmeAlignModelData:ModelDataType

{

name = "intern.KD.adel.hsa.model\_data",

type\_name = "KDxADELxHSAxMODELDATA:StrategyModelResults",

data\_category = "output\_report\_external\_control"

},

typedef struct

{

int waferId

<microhelp = "Identification of the wafer">;

ADELov:StrategyIDType strategyId

<microhelp = "Identification of the alignment strategy">;

**ADELmeAlignModelData:SaModelData** saModelData

<microhelp = "Stage alignment model data">;

} KDxADELxHSAxMODELDATA:StrategyModelResults

<microhelp = "Strategy model results for stage alignment">;

**EXTIF Changes needed**

* VP item (INFORMPRO2 / EXT\_IFWA)
* Descriptions (mail Martien), see patch
* Version number?
* DP (see below)

**DP**

* Remove ADEL part in model\_data ?
* Use only has.
* "/vobs/vb322/40224348231/KD/com/int/bin"
* "/vobs/vb322/40224348231/KD/com/int/inc"
* "/vobs/vb322/40224348231/KD/com/int/lib" "/vobs/vb322/40224348231/KD/com/int/bin/KD\_FIWA\_fine\_wafer\_align.c"
* "/vobs/vb322/40224348231/KD/com/int/bin/KD\_LO\_lot.c" **"/vobs/vb322/40224348231/KD/com/int/bin/makefile"**
* "/vobs/vb322/40224348231/KD/com/int/bin/KD\_CSA\_combined\_stage\_alignment.c" "/vobs/vb322/40224348231/KD/com/int/bin/KD\_PSA\_paris\_stage\_alignment.c"
* "/vobs/vb322/40224348231/KD/com/int/bin/KD\_PSM\_plate\_shape\_measurement.c"
* "/vobs/vb322/40224348231/KD/com/int/lib/makefile"
* **"/vobs/vb322/40224348231/KD/com/ext/ddf/ADELmeAlignModelData.ddf" "/vobs/vb322/40224348231/KD/com/int/bin/KDAL\_ADEL\_model\_data.c"**
* **"/vobs/vb322/40224348231/KD/com/int/inc/KDAL\_ADEL\_model\_data.h"**
* **"/vobs/vb322/40224348231/KD/com/int/ddf/KDAL.ddf"**
* "/vobs/vb322/40224348231/KD/com/int/tst/KDtest\_model\_data\_parameters.c" "/vobs/vb322/40224348231/KD/com/int/tst/KDtest\_mark\_acceptance.c"
* "/vobs/vb322/40224348231/KD/com/int/tst/KDtest\_mark\_rejects.c" "/vobs/vb322/40224348231/KD/com/ext/ddf/KDxADELxHSAxMODELDATA.ddf"
* "/vobs/vb322/40224348231/KD/com/int/tst/KDtest\_model\_data\_instance.c"
* "/vobs/vb322/40224348231/KD/com/int/tst/KDtest\_model\_data\_stage\_alignment.c"

{

name = "intern.KD.adel.hsa.model\_data",

type\_name = "KDxADELxHSAxMODELDATA:StrategyModelResults",

data\_category = "output\_report\_external\_control"

},