Jet Propulsion Laboratory
California Institute of Technology

Update for SITs on CGI processing and wavefront sensing and control baseline and plans

Eric Cady
Jet Propulsion Laboratory, California Institute of Technology

12/12/19

© 2019. California Institute of Technology. Government sponsorship acknowledged.

Overview

CGI is still assembling an official policy on accepting algorithm contributions. We will present it at a future Stanford meeting.

For benefit of interested parties, we cover the following items today, related to planned CGI capabilities:

- **Review of CGI hardware baseline**
- **Review of CGI telemetry baseline and data rate**
- **Review of CGI high-order WFSC software baseline and schedule**
- **Path to updates of CGI behaviors**

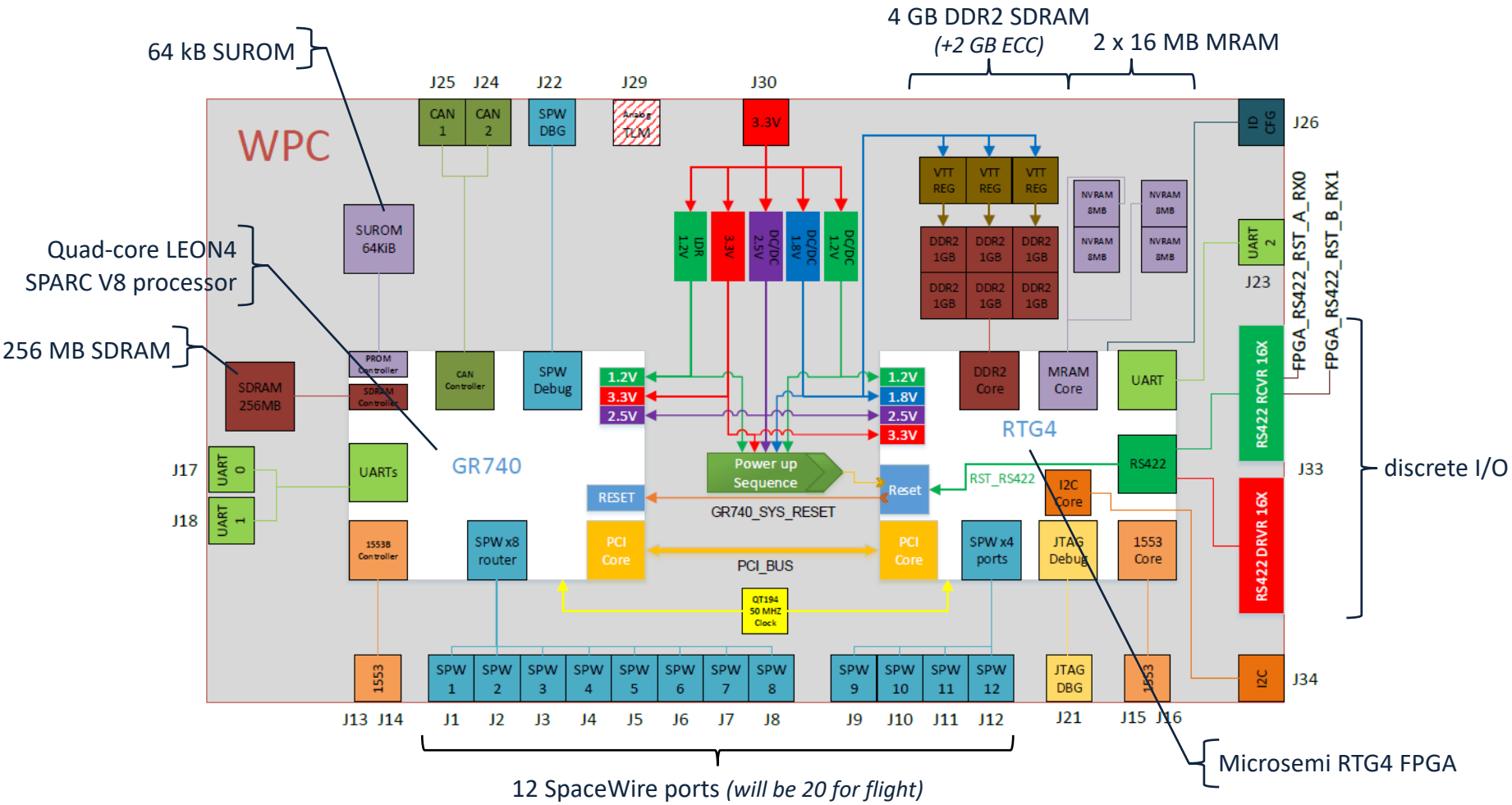
CGI processing hardware: WPC

CGI has two processing cards: ICDH and SSP

- **Both are copies of WFIRST Processing Card (WPC), which is GSFC-delivered and is replicated across WFI and the spacecraft**

- **WPC has:**
  - A GR740 LEON4 processor from Cobham-Gaisler
    - Rad-hard, 32-bit architecture, 4 cores @ 250MHz apiece
    - LINPACK puts it at 23Mflop per core for matrix operations
    - 256MB of DDR SDRAM attached locally
  - An RTG4 FPGA from Microsemi
    - Provides access to 4GB of error-corrected DDR SDRAM for large volatile memory needs
    - Provides access to 32MB of MRAM for flight software storage
  - A 33 MHz PCI bus between the two
  - Up to 20 Spacewire ports (along with several others we may or may not use, e.g. RS-422 to power slice)

WPC



Goddard-supplied WFIRST Processor Card is the platform for the SSP slice

CGI processing hardware: SSP + ICDH

SSP is a dedicated processor for high-order wavefront sensing and control.

- **Runs computationally intensive algorithms without the ICDH multitasking concerns**
  - Message-passing between SSP and ICDH will be used to get information to and from cameras and mechanisms
  - Independent connection to CGI solid-state recorder (SSR) so transfer of large stored data items does not bog down ICDH

- **Will use remaining RTG4 resources for "Fourier Optics Engine"**
  - FPGA-based single-precision 2D Fourier transform and complex elementwise mask multiply
  - Can do full FT and multiply in < 0.252 for matrices of any size up to 1739x1739
  - Can be chained together to do one operation after another, up to 9 (TBR)
  - Used primarily for calculations for high-order wavefront control, but also usable for other operations requiring Fourier transform capability (use a mask of all ones for FT only). Available but not a requirement.

ICDH handles interfaces with spacecraft and with electronics controlling mechanisms and cameras. It will pass messages and telemetry, and run small control loops.

- **Not suitable for heavy computation; it has too many other responsibilities**

CGI processing hardware: other

SSR will be a solid-state recorder of at least 60GB

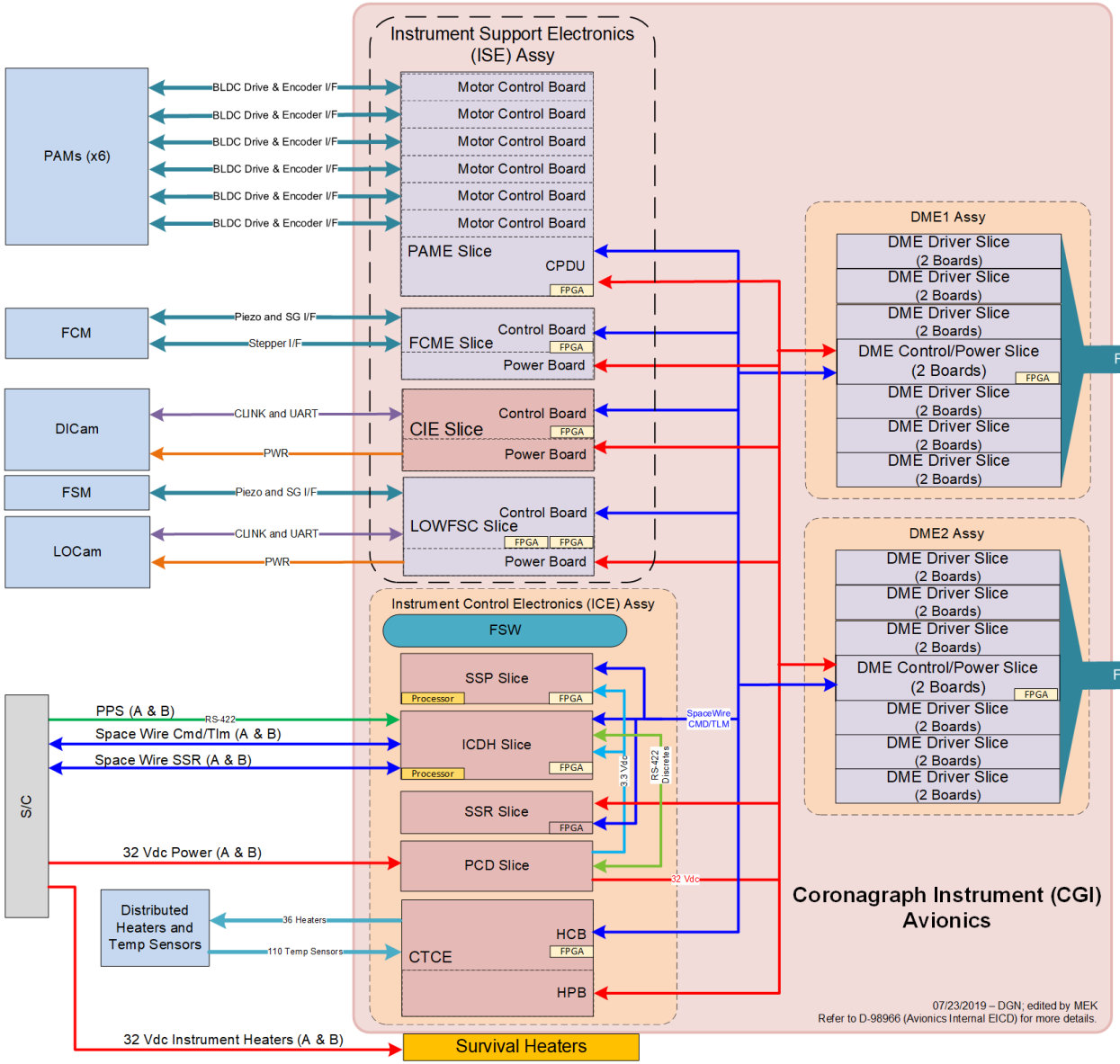
- Suitable for long-term storage of large data items (calibration data, wavefront control Jacobians, etc.)
- Will have dual-port access to ICDH, so either can talk to it independently
- Finite (but large) number of writes permitted
- To be purchased from vendor

CGI will also have several other JPL-produced "slices", boards dedicated for control of particular hardware:

- PCD for power
- CTCE for thermal
- PAME for Precision Alignment Mechanisms (PAMs), the x-y stages holding masks
- LOWFSC slice for the Fast Steering Mirror (FSM), the low-order camera (LOCAM), and all the low-order estimation and control between those two.
- FCME for the Focus Control Mechanism (FCM)
- DME slices for deformable mirror (DM) control
- CIE for interfacing with the EXoplanetary systems CAMera (EXCAM)

Both cameras also have vendor-sourced proximity electronics directly connected to the camera itself.

Avionics block diagram



Legend:

- Tested together before delivery to ETB and II&T
- Delivered separately to users before integration in ETB and II&T

List of Abbreviations:

- CIE = Camera Interface Electronics
- CTCE = Coronagraph Thermal Control Electronics
- CPDU = Command & Power Distribution Unit
- DICam = Direct Imaging Camera
- DM = Deformable Mirror
- DME = Deformable Mirror Electronics
- FCM = Focus Control Mechanism
- FCME = Focus Control Mechanism Electronics
- FSM = Fast Steering Mirror
- FSW = Flight Software
- HCB = Heater Control Board
- HPB = Heater Power Board
- ICDH = Instrument Command & Data Handling (GSFC WPC)
- LOCam = Low-Order Wavefront Sensor Camera
- LOWFSC = Low Order Wavefront Sensing & Control
- PAM = Precision Alignment Mechanism
- PAME = Precision Alignment Mechanisms Electronics
- PCD = Power Conditioning & Distribution
- SSP = Starlight Suppression Processor (GSFC WPC)
- SSR = Solid State Recorder

CGI telemetry

CGI will send down almost every piece of data it collects on the high-rate link:

- **All EXCAM images**
- **All commanded DM settings (both high-level WFSC-commanded and 16-bit exact DAC settings)**
- **All final encoder positions of mechanisms after moves**
- **All calibration products**
- **All electric field estimates**
- **All LOWFS estimator products being sent to any of the low-order control loops (line-of-sight, focus, Zernike)**
- **All spacecraft reaction-wheel speed and torque data received from the WFIRST ACS to CGI for line-of-sight control**
- **Response codes from every completed algorithm**
- **A regular telemetry output of all temperatures and voltages within CGI**
- **All LOCAM images, summed up to once every 10 seconds**
  - If we'd output the raw 1kHz output, it would driven our implementation and busted our data rate requirements.

These should be available no later than 72 hours after collection.

A subset of this will be sent on the housekeeping link for near-real-time observer access.

CGI data rate

L3 CGI - CGI Maximum High Rate Data Bus Data Volume

The CGI shall not transfer more than 3 Tbits over any 24 hour period to the spacecraft on the high rate data bus.

L3 CGI - CGI Maximum Data Rate Allocation

The CGI shall remain within a maximum data rate to the SSR of 60Mbps.

	Requirement	CBE	Margin
Data Rate	60 Mbps	10.2 Mbps	83%
Data Volume	3 Terabits/day	0.88 Terabits/day	71%

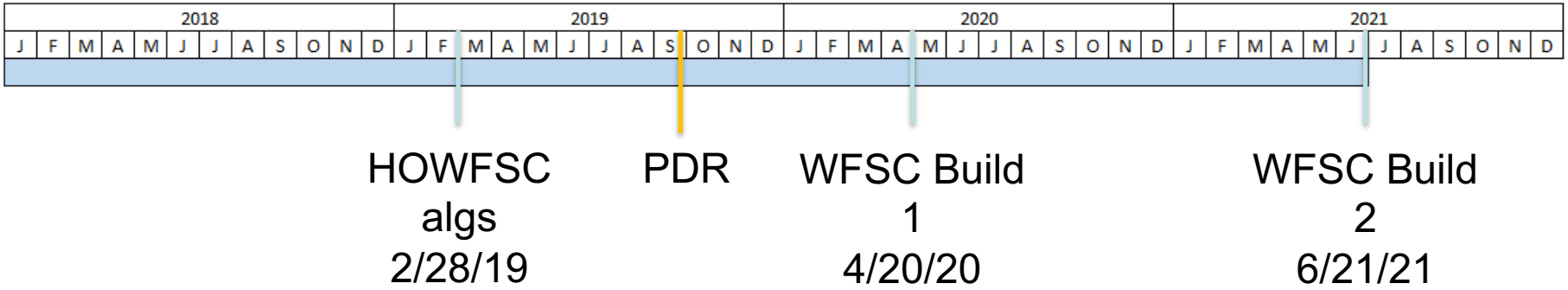
Some room on data volume if needed.

CGI HOWFSC software scope

Current HOWFSC baseline:

- **Pairwise probing for wavefront estimation**
  - Pairs of DM settings modulate the starlight, and images of these are combined with a system model to extract amplitude and phase
- **Electric-field conjugation for wavefront control**
  - Basically a weighted least-squares fit, with regularization
    - Uses a DM Jacobian, a model-based representation of the change in electric field from moving each actuator
    - Do an Ax = b solve to back out the next DM setting given the estimated E-field
    - Many parameters to tune it (weights, regularization schedules vs. iteration, size of region being controlled…)
  - Requires careful precalibration to avoid model mismatch
- **Phase retrieval for wavefront estimation during precalibration**
  - Use 6 lenses on DPAM (imaging lens, pupil lens, 4 weak lenses) to introduce diversity
  - Estimates pupil-plane phase using parametric fitting for low-order aberrations and modified Gerchberg-Saxton for higher-spatial-frequency aberrations
- **Other algorithms for cleaning images, aligning masks, doing WFSC astrometry/photometry/plate scale calcs…**

CGI HOWFSC software schedule



Next delivery: end-to-end WFSC in April 2020 to FSW

- All remaining LOWFS and calibration
- Any HOWFSC updates since Feb 2019
- Very few missing pieces at present; biggest gaps are automation of calibration routines which were previously operator-in-the-loop
- Focus is on delivering a complete, working baseline

Build 2 (June 2021):

- incorporate any feedback from FSW implementation of Build 1
- Possible opportunity for algorithm updates with some WFSC support

After Build 2: WFSC will be wrapped up and personnel moved elsewhere within CGI (I&T, Testbed, SE, …); supporting external contributions will be harder

Path to CGI behavior updates

Four paths to updating CGI behavior (from easiest to hardest):

1. **Re-ordered commands**
   - CGI will have a command dictionary which gives a list of all knobs available for an operator to turn
     - Most will be low-level hooks to move mechanisms, open/close loops, take images, etc.
     - A handful will be "flight software behaviors", which string together low-level behaviors with processing activities and data passing to create integrated functions which can also be called as commands. Examples:
       - align a pupil-plane mask
       - compute plate scale for one or more filters
       - run the CGI baseline HOWFSC loop
   - CGI will operate on sequences of commands strung together
     - Very limited ability to pass information between them, however
   - If a new CGI behavior can be assembled from a sequence of existing commands, it could be accommodated with essentially no extra work
     - Example: rotating satellite spots during a blind-search observation, to keep an star-centration reference while providing full coverage

Path to CGI behavior updates

Four paths to updating CGI behavior (from easiest to hardest):

2. **Updating parameters and stored data**
   - CGI will use stored configuration data to define parameters for calibration and wavefront control routines
     - WFSC-relevant examples:
       - regularization parameter schedules
       - per-pixel and per-actuator weighting matrices
       - region of interest for control ("dark hole" location)
       - DM probe shapes, locations, and magnitudes
       - combinations of subfilter and star off-axis offset to use for control ("control channels")
     - Other parameters include control-loop parameters, alignment tolerances for iterative routines, etc. Can include parameters in FPGAs.
   - Behavior changes that can be done by updating parameters within the existing CGI framework are the next easiest to do
     - Example: changing the control region of interest to dig a smaller (but ideally deeper) dark hole on one side of the PSF
     - Example: changing the regularization schedule to correct more aggressively more often if the calibration supports it

Path to CGI behavior updates

Four paths to updating CGI behavior (from easiest to hardest):

3. **Updating flight software (FSW)**
   - JPL Design Principles 4.1.7.3: "**Capability to modify flight software** - The design shall include the capability to modify flight software of spacecraft components containing reprogrammable flight software in-flight through the C&DH subsystem via the uplink command function, or prior to launch via the umbilical link."
     - At JPL, Design Principles (with "shall") must be adhered to unless a waiver is explicitly granted. We are not planning a waiver on this that I am aware of.
     - This only applies to "reprogrammable" flight software, so SSP/ICDH yes, FPGAs no.
   - So yes, we can update any of the CGI FSW at any point, even after launch. But the procedure to do so is much more involved than the first two.
     - Expect it to be a much slower process (see next slide)…

Path to CGI behavior updates

- **Code must be written (C/C++ for VxWorks) with tests**
- **Code must go through software quality checkout and regression testing, and be tested on the ground-based instrument testbed to verify it**
  1. works as expected
  2. doesn't break the instrument
  3. doesn't break any of the existing software functionality
- **Documentation must be updated (e.g. command dictionary, design doc, user guides)**
- **There will be a Software Review and Certification Record (SRCR); this review must be completed to transfer the code from the development team to the operations team who will upload it**
- **If the FSW change is introducing a new command, Ground System and possibly spacecraft software must be updated as well to handle the new command, with their own testing**
- **And anecdotally, expect to meet resistance from many stakeholders**
  – A FSW update could brick an otherwise working instrument, and nobody wants a piece of that one…

<u>None of these are impossible to overcome</u>, but be aware it will cost time and money to get through them all

Path to CGI behavior updates

Four paths to updating CGI behavior (from easiest to hardest):

4. **Updating FPGA behavior/electrical-hardware configuration**
   - Not possible after launch
   - Nearly impossible after start of II&T (enormous standing-army cost, likely only to happen if something is genuinely broken and not caught earlier)
   - Should not really happen after Critical Design Review (CDR) in Nov. 2020
     - Presenting detailed design and design validation at this gate, ready to start implementation
   - For existing FPGAs, will be limited by the needs of other CGI activities that already use the resources on a part
   - Will be competing for resources (budget, engineer time) with existing CGI activities
   - Technically possible, but really very disfavored unless it's an immediate change that helps CGI meet its existing requirements with more margin or lower cost