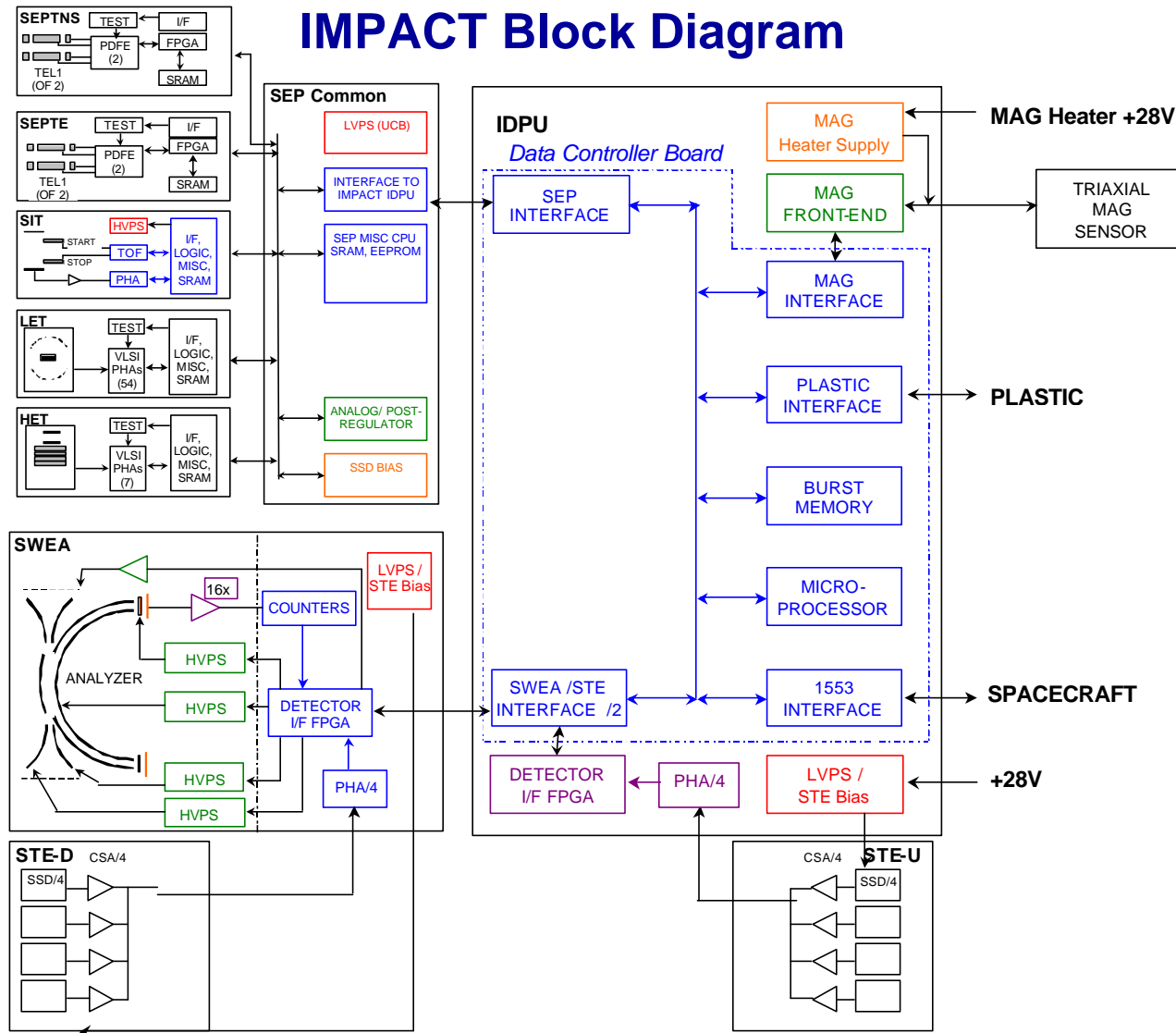


IDPU Overview

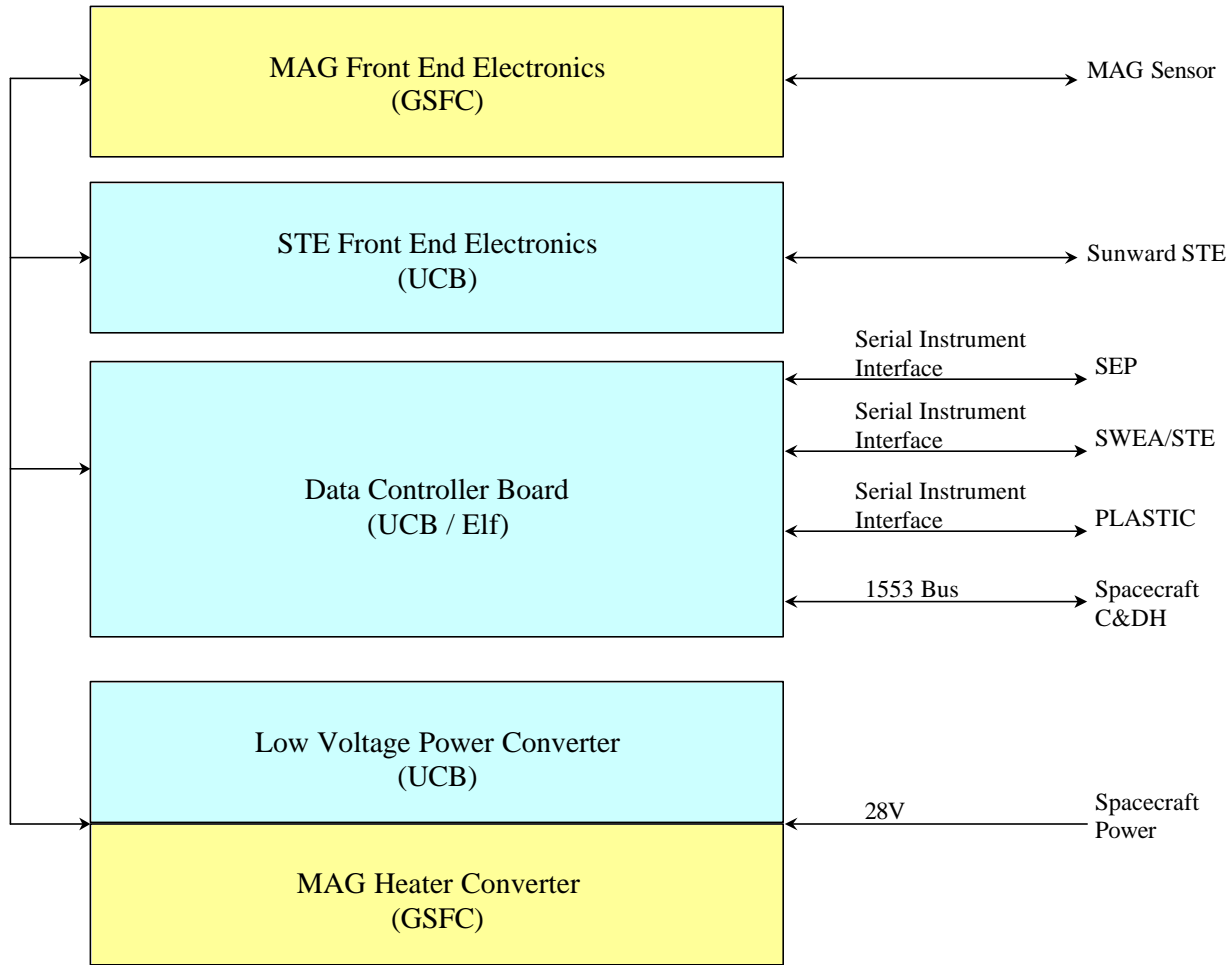


STEREO IMPACT

IMPACT Team Meeting
2001-December-14

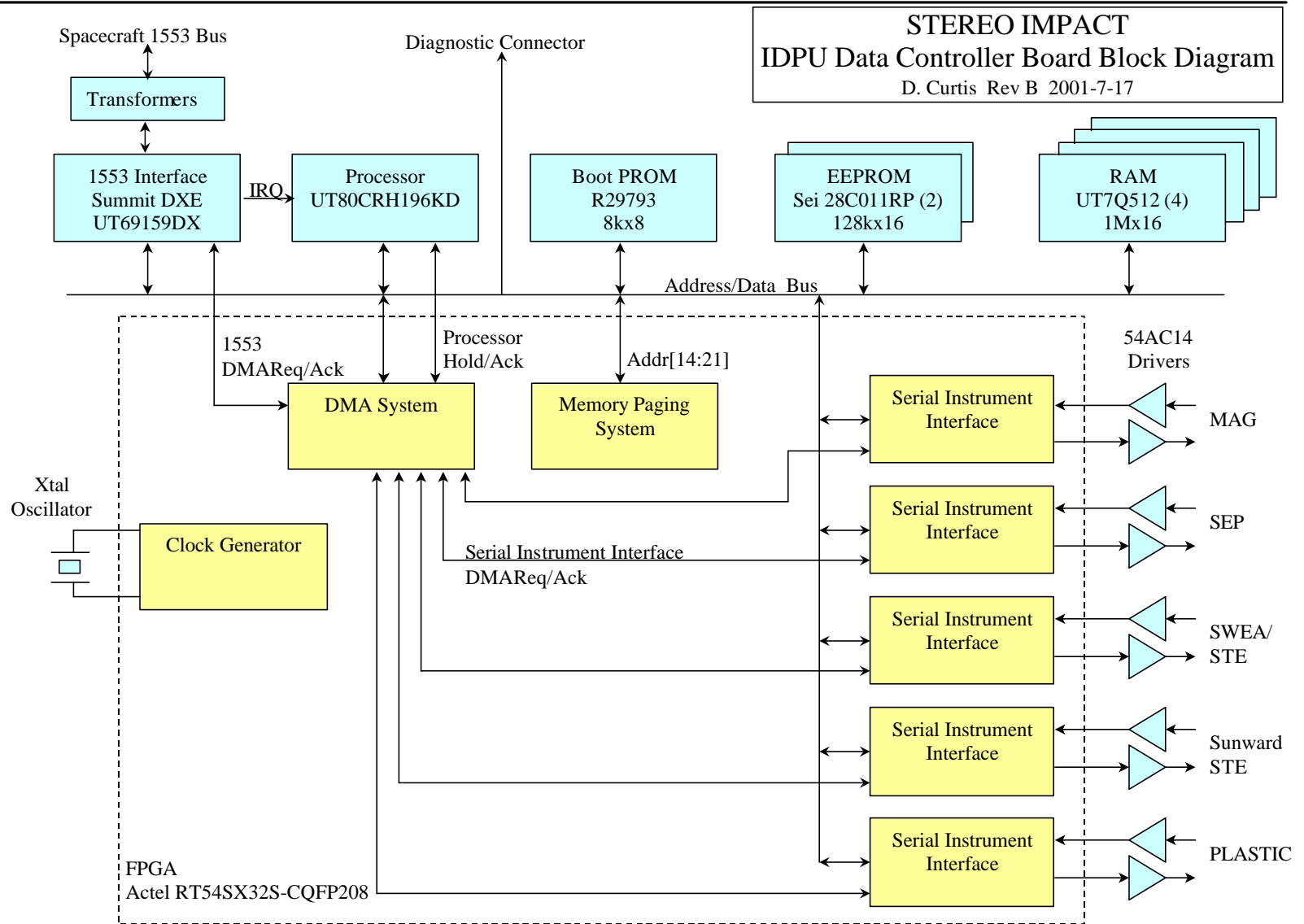
"Stackable Connector" Interconnect:
+/-12V, +/-5, +5V, +2.5V
MAG Serial Instrument Interface
MAG Heater Control
STE Serial Instrument Interface

STEREO IMPACT
IDPU Block Diagram
D. Curtis Rev B 2001-7-17



STEREO IMPACT

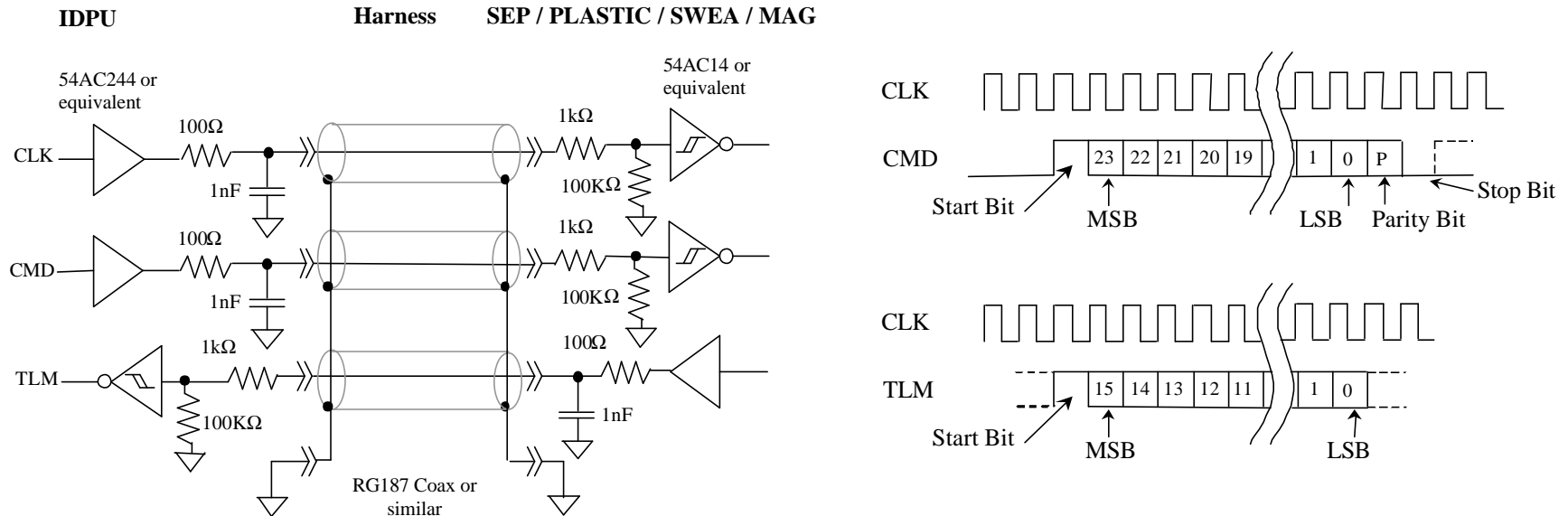
IMPACT Team Meeting
2001-December-14



IDPU Hardware Features

- **16-bit processor with high speed multiplier**
 - required for moment calculations
- **8Kbyte boot PROM**
- **256Kbyte EEPROM for code and tables**
- **2Mbytes RAM**
 - Mostly for Burst memory, plus data buffering
- **Watchdog Timer**
- **1553 Interface**
 - Message transfer into processor memory buffer automated by 1553 hardware
 - Processor interrupt on error or completed message transfer
- **Sample clock interrupt (256Hz, TBR)**
 - Used for timekeeping, time-critical tasks
- **Serial Instrument Interfaces**
 - Automatic transfer of instrument data into message FIFOs
- **Memory paging**
 - maps physical memory into 64K processor addressable memory space in 4 16kbyte pages

Serial Instrument Interface



- **Commands include 8 bit destination code and 16 bits of data**
- **Telemetry messages consist of 16-bit words bounded 17 zeros**
 - **Up to 1025 words per message**
 - **The first word of the message includes a message type code and a message length count**
- **No handshaking is used**

IDPU Software Requirements

IDPU Software Requirements

- **First draft of Software Requirements Document 6/01**
 - Rev B, October/01
- **First draft of PLASTIC software requirements from UNH 6/01**
 - Rev B, November/01
- **Top level requirements:**
 - Support Spacecraft 1553 Interface per the Instrument ICD
 - Support Instrument Interfaces per the hardware and software specification
 - Collect, compress, and format telemetry data into CCSDS packets
 - Pass on mode-setting commands from the ground to instruments
 - Limited automation:
 - Set SWEA offset voltage to track spacecraft bias voltage
 - PLASTIC solar wind tracking and entrance system selection
 - MAG ranging
 - Instrument safing (response to HV anomaly, and spacecraft thruster/power down notifications)
 - Support a Burst memory system to collect high time resolution data for short intervals based on a burst trigger criteria
 - Share burst trigger data with SWAVES via RT to RT 1553 transfers

1553 Interface Requirements

- **Support the interface protocol as described in the ICD**
 - Receive, decode, and forward command packets
 - Send telemetry packets
 - Normal telemetry, Beacon telemetry, Housekeeping telemetry
 - Receive, decode, and synchronize local clock to UTC time code
 - Receive and act on spacecraft status
 - Power down warning; safe instruments
 - Thruster firing warning; safe instruments
 - SSR status – send only state of health when SSR partition almost full
 - Send instrument status
 - Power-down request (on fault detection)
 - Respond to Loopback, TBD mode codes
- **Support 1553 Interface chip**
 - Setup & maintenance
 - Interrupt on message in/out
 - Error response

MAG Software Requirements

- **Average MAG data to 4 samples/second, format into CCSDS packets**
- **Format 32 sample/second data into CCSDS packets for Burst memory**
- **Average MAG data to 1 minute and pass to Beacon telemetry formatter**
- **Perform ranging functions to select one of two gains based on previous measurements (only expected to be used on the ground)**

SEP Software Requirements

- **Pass on all SEP command packets (by ApID) and Spacecraft Time via Serial Instrument Interface**
- **Pass on all Telemetry Packets received via Serial Instrument Interface to Telemetry packet queue**
- **Pass SEP Housekeeping data received via the Serial Instrument Interface to Housekeeping packet formatter**
- **Pass SEP Beacon data received via the Serial Instrument Interface to Beacon packet formatter**

STE Software Requirements

- **Initialize and periodically reload STE Energy to Accumulator look-up table via Serial Instrument Interface from IDPU EEPROM**
- **Initialize and periodically reload STE threshold DAC values via Serial Instrument Interface from IDPU EEPROM**
- **Log-compress counters and format into CCSDS packets to pass to the Burst system**
- **Time Average and Log Compress counters and format into CCSDS packets for real-time telemetry**
- **Time and Space Average and Log Compress counters and pass to Beacon telemetry formatter**

SWEA Software Requirements

- **Initialize and periodically reload SWEA voltage waveform look-up table via Serial Instrument Interface from IDPU EEPROM**
- **Initialize and periodically reload SWEA control registers via Serial Instrument Interface from IDPU EEPROM**
- **Accumulate raw counter measurements into a three-dimensional distribution measurement**
- **Compute Moments, Pitch Angle Distributions and Full 3D distributions with desired time and space resolution and format into CCSDS packets for the Burst, Real Time, and Beacon telemetry streams**
- **Adjust SWEA bias voltage based on measured distribution function to offset spacecraft charging effects**

PLASTIC Software Requirements

- **Instrument Control Tasks:**
 - Decoding and routing instrument mode and table load commands
 - Controlling the energy at which apertures are switched based on measured count rates
 - Controlling “Solar Wind Tracking Mode” when Bursts are triggered to get high time resolution data
 - HV safing in response to an Arc
- **Data Tasks: Collect, compress, and format**
 - Housekeeping data
 - Monitor Rate data
 - Matrix data (from the classification board)
 - Proton and Alpha Moments
 - Reduced Proton and Alpha Distributions
 - Heavy Ion Distributions
 - Raw Event Data
 - Event Prioritization
 - Beacon Mode data

Burst System Requirements

- **High time resolution sampling of selected data**
 - **MAG @ 32Hz**
 - **SWEA, STE at 2 seconds**
 - **PLASTIC H,He at 4 seconds (TBR)**
- **Continuous data collection into a circulating buffer**
 - **PLASTIC burst mode must be triggered, and so is not continuously recorded**
 - **Buffer sized for on the order of 10 minutes of high time resolution data**
- **Continuous evaluation of burst trigger criteria based on instrument data**
 - **Trigger on changes in SWEA count rate in a selected energy band**
 - **Trigger cooperatively with SWAVES based on SWAVES activity**
- **Freeze collection buffer a fixed interval after the trigger event to provide data both before and after the trigger**
- **Continue to look for more triggers, saving the “best” event in the time interval between burst playback**
 - **Play back burst data using a fraction of the normal telemetry on a continuous basis.**

IDPU Software Development Plan

Software Development Plan

- **First draft Software Development Plan 8/01**
- **IDPU Flight Software to be developed at UCB by a single programmer**
 - Backup plan for an additional programmer
- **Code to be developed in modular, structured assembly code**
- **Software to be developed and tested on the ETU IDPU, with ETU Instruments or simulators**
- **Software to be tested at the module level, then at the IDPU level in an acceptance test, and finally at the system level in interface and suite tests**
- **Software to be developed in builds of progressing complexity:**
 - First Build to test processor hardware
 - Second build provides at least minimal functionality required to verify suite hardware (without automation and more complex telemetry products)
 - Third build should be for Flight
- **Prior to installation in the flight hardware, code is maintained using a log book which contains the development history, testing, problems, etc.**
- **Following installation in flight hardware, software is under the same control as the hardware (configuration control, problem reports, etc.)**

Software Development Tools

- **Use Commercial Phyton PC-based 80C196 development tools**
 - Assembler/Linker
 - Simulator (for early module testing)
- **ROM Emulator (for fast testing)**
- **Logic analyzer (for those subtle timing problems)**
- **ETU Data Controller Board**
 - Note that IDPU Simulator GSE is based on the same architecture as IDPU DCB (commercial 80C196KD with similar memory and FPGA); can be used for early software development and test
- **EGSE:**
 - **APL Spacecraft Emulator**
 - UCB-developed Command and Telemetry GSE
 - UCB-developed Science Display GSE (MAG,SWEA,STE)
 - SEP, PLASTIC Science Display GSE
 - UCB-developed Instrument Simulator
 - Instrument ETUs
 - UNH-provided PLASTIC simulator

Software Development Schedule

- **Software Development Plan** PDR (9/01)
- **Software Requirements Document(s)** PDR (9/01)
- **Software Requirements/Design Review** 12/01
- **Software Design Complete, start Coding** 1/02
- **First Build Complete, ETU Available** 8/02
- **Software Design Review (part of CDR)** CDR (11/02)
- **Critical Code Walkthrough /
Acceptance Test Review** 6/03
- **Boot PROM Acceptance Test** 8/03
- **Boot PROM Install in DCB of FM1** 9/03
- **Second Build Complete** 9/03
- **Third Build Complete, Acceptance Test** 3/04

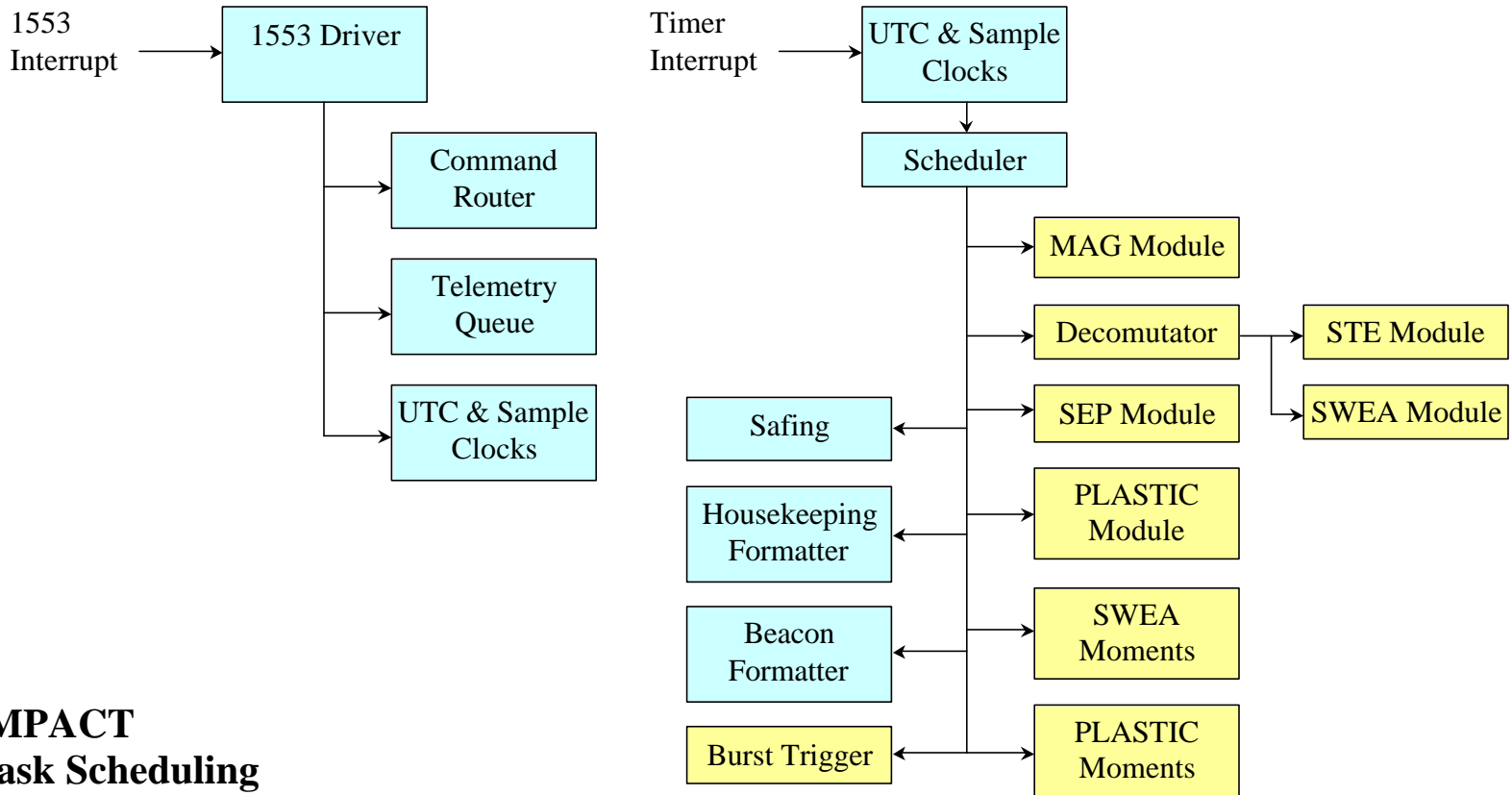
IDPU Software Design

IDPU Software Architecture

- **Simple Polled-Loop operating system**
 - 1553 and 256Hz interrupts take care of time-critical functions
 - Input data FIFOs hold data until the relevant task comes to collect it
 - The majority of the tasks involve short intervals of processor to deal with data, so the polling loop frequency is expected to be much greater than 1Hz, providing adequate response time to all tasks
 - The few events that require better than 1 second latency (such as the PLASTIC aperture select) will be taken care of in the interrupt handlers
 - A few tasks run infrequently require more processing (such as the PLASTIC matrix data formatting) and will be broken up to level the processing load
 - A somewhat more complex priority-based time-slice operating system has been considered, but is currently more complex than is required
- **Tasks are modular and as independent as possible**
 - Especially PLASTIC, to allow independent operation by UNH
- **Hardware designed to relieve software of most time-sensitive tasks**
 - Instrument sequencers allow instruments to cycle independently
 - DMA system collects data into IDPU memory automatically
- **Code will utilize software watchdogs on each task, together with a hardware watchdog, to monitor operation**
- **Software will regularly re-load all instrument registers to protect against SEU**

Boot Sequence

- **System boots from PROM**
- **PROM verifies the EEPROM code image checksum**
 - **On bad checksum, system waits in PROM for new code uplink from the ground via spacecraft 1553 interface**
- **PROM waits a few seconds for a possible abort message prior to transferring control to EEPROM**
- **EEPROM initializes each task and then starts task scheduler**
- **EEPROM can be reloaded or patched from BOOT PROM**



**IMPACT
Task Scheduling
Diagram**

1553 Handler

- **Invoked by interrupt from 1553 Interface**
 - Completed message transfer
- **Telemetry Message:**
 - Fetch a new packet pointer from telemetry queue (if available) and setup 1553 for next transfer
- **Command Message:**
 - Command Router extract command packets
 - Pass packets to tasks based on command ApID
- **Timing/Status Message (1/sec):**
 - Update UTC clock
 - Invoke command scripts on events (instrument power-down or thruster firing)
- **SWAVES Interface (1/sec):**
 - Pass SWAVES data to Burst Trigger Task
 - Set up to send IMPACT Burst Trigger data to SWAVES

Telemetry Queue

- **Accepts CCSDS telemetry packets from various tasks**
 - **MAG**
 - **SWEA**
 - **STE**
 - **SEP**
 - **PLASTIC**
 - **Housekeeping**
 - **Beacon**
 - **Burst**
- **Provides packets in a FIFO mode to 1553 handler**
- **Tasks are expected to generate telemetry at their allocated rate**
 - **Some subsystems, such as housekeeping, will have programmable telemetry rates. The IMPACT Team must coordinate rate allocations between tasks.**
- **A bit-counting scheme is used to match packet generation rate to allocation in order to avoid over-filling the IMPACT/PLASTIC SSR allocations**
- **Any telemetry allocation not used by the instruments will be used to play out the Burst Memory**
- **Command Response and Error Message packets generated asynchronously**

Housekeeping & Beacon Modules

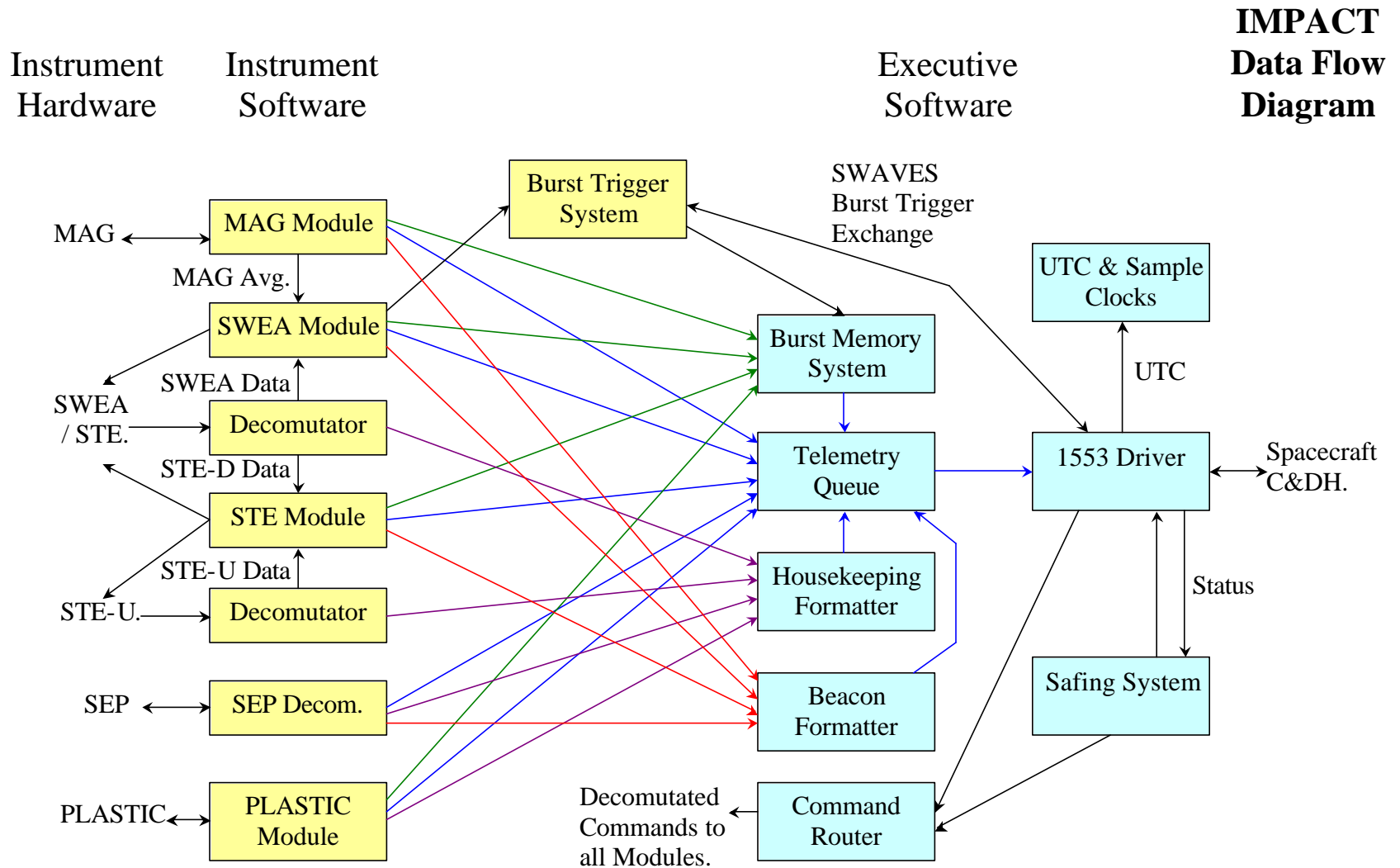
- **Tasks provide housekeeping data to housekeeping formatter**
 - Each task has a fixed allocation in the housekeeping packet
- **Housekeeping packet rate is nominally once a minute**
 - Can be increased during I&T and commissioning to improve test sequence speed
- **Tasks provide Beacon data to Beacon formatter**
 - Each task has a fixed allocation in the housekeeping packet
- **Beacon Packets generated once a minute each for PLASTIC and IMPACT**

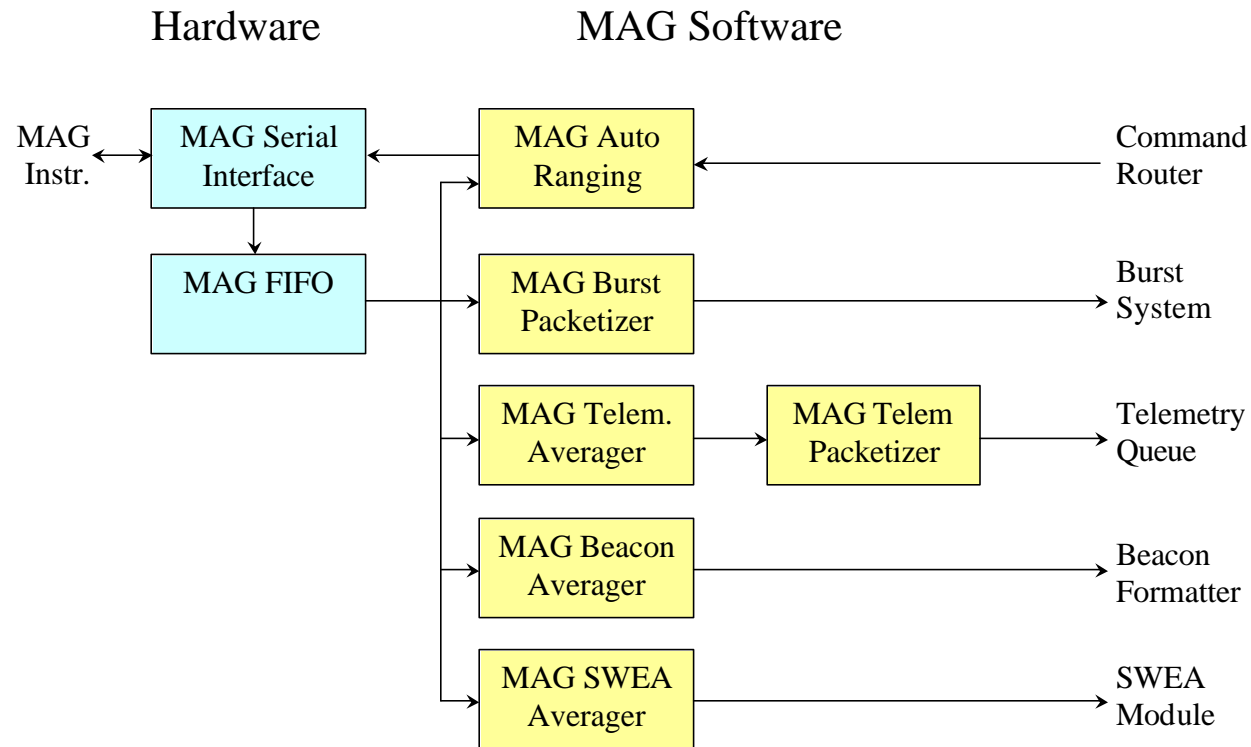
Safing System

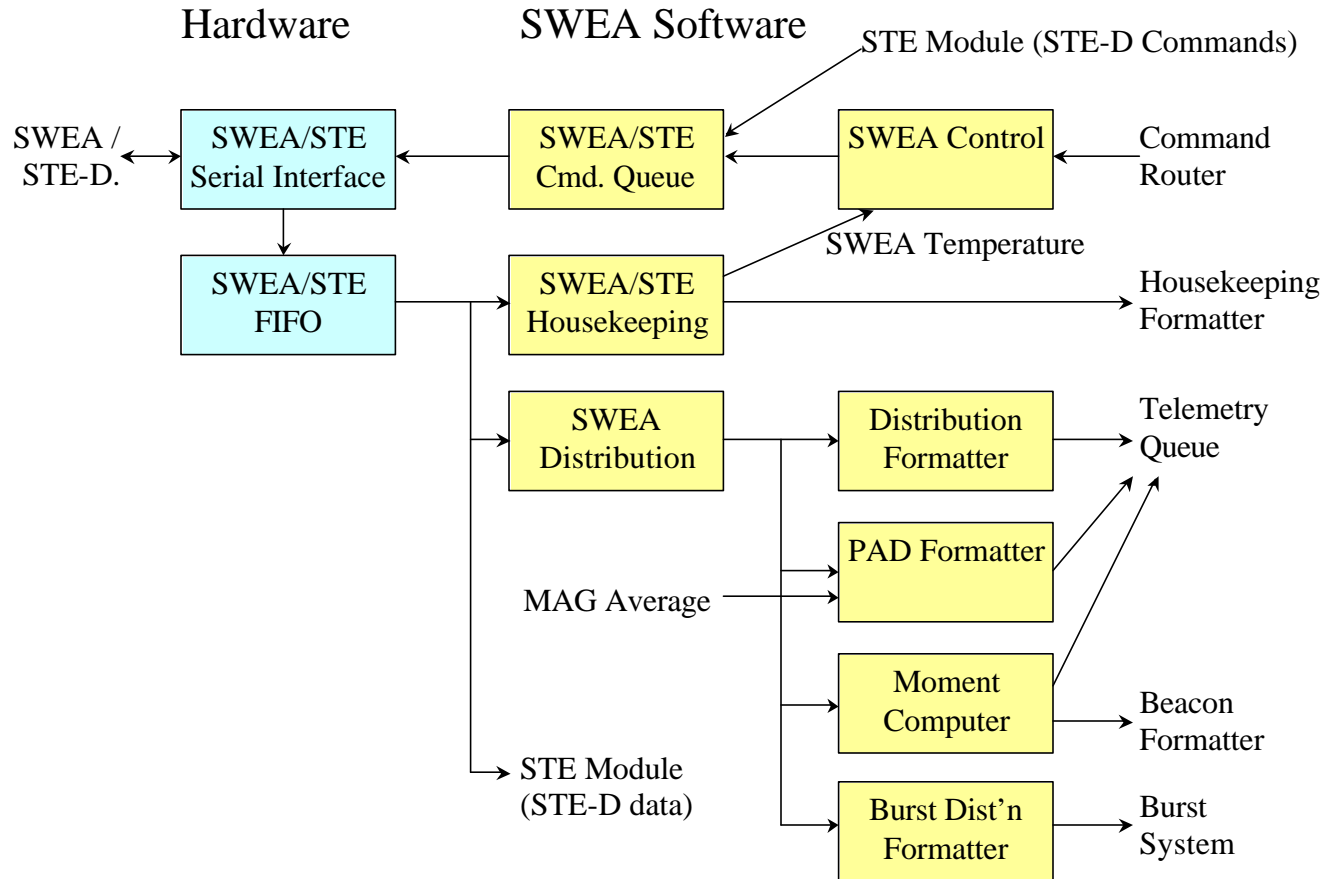
- **This system is used to invoke a sequence of operations to be performed in response to an event**
- **The operations are executed at a fixed rate**
- **Operations are in the form of command packets which are distributed just as if they were received over the 1553 interface**
- **Operation sequences are programmable from the ground to allow flexibility**
- **Operation Sequences can be invoked by tasks or by command**
- **Currently planned sequences:**
 - **Power Down Preparation (1553 warning)**
 - **Thruster Firing Preparation (1553 Warning)**
 - **End of Thruster Firing (1553 Warning)**
 - **Fine Pointing Stats (15530)**
 - **PLASTIC HV Discharge**
 - **Temperature / Voltage / Current exceedences**

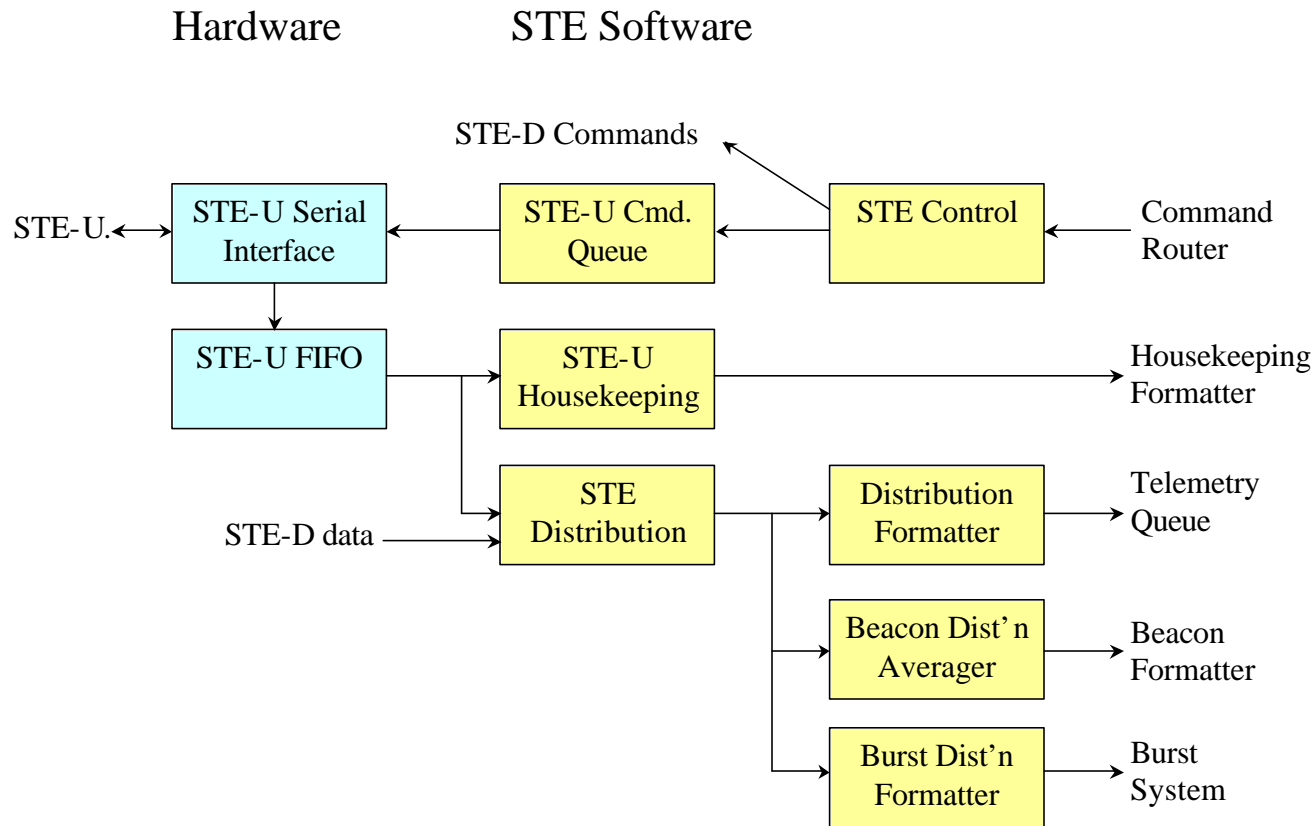
Burst System

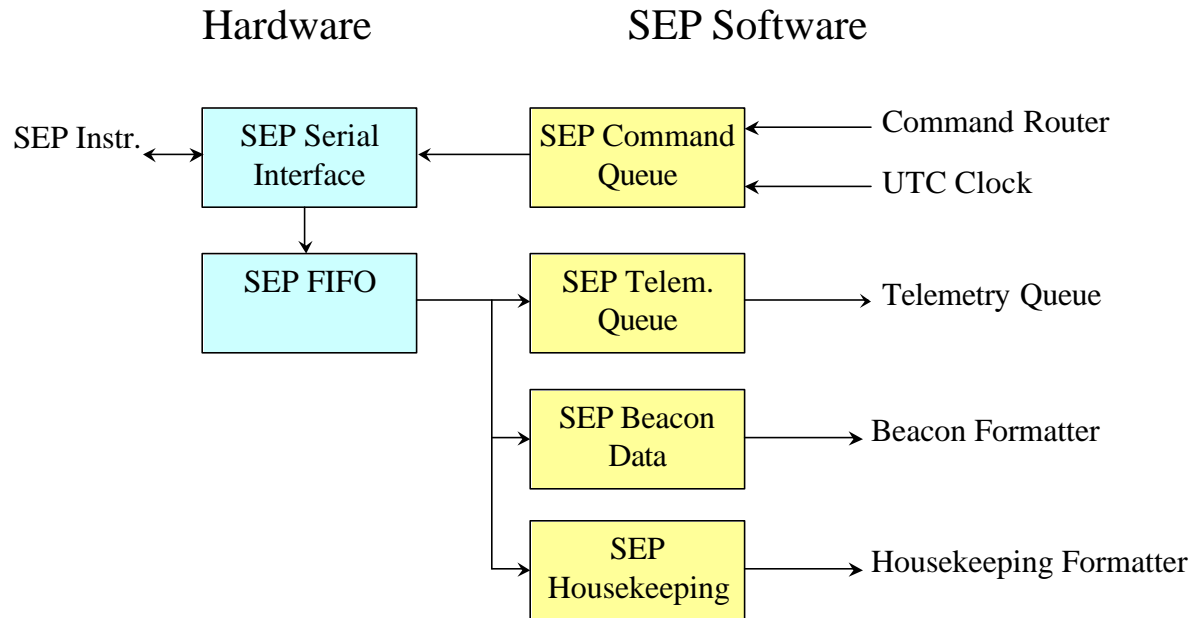
- **Tasks provide a continuous stream of packets into the Burst system**
 - **MAG, SWEA, STE, PLASTIC**
- **The Burst system records the data into a circular buffer in RAM**
 - **This allows pre-trigger data to be saved**
 - **Three buffers are allocated in RAM: Transmitting, Best, and Current**
- **A trigger criteria system evaluates data from SWEA and SWAVES to detect an event**
- **If a new event has higher criteria than the previously recorded “Best” event, the current event collection is completed, and the new buffer is saved as “Best”**
 - **The old “Best” buffer now becomes “Current” and is over-written with the continuous telemetry stream, looking for an even better burst**
- **This continues until the “Transmitting” buffer has been read out to telemetry. The “Best” buffer now becomes “Transmitting”, and the old “Transmitting” becomes “Current”.**

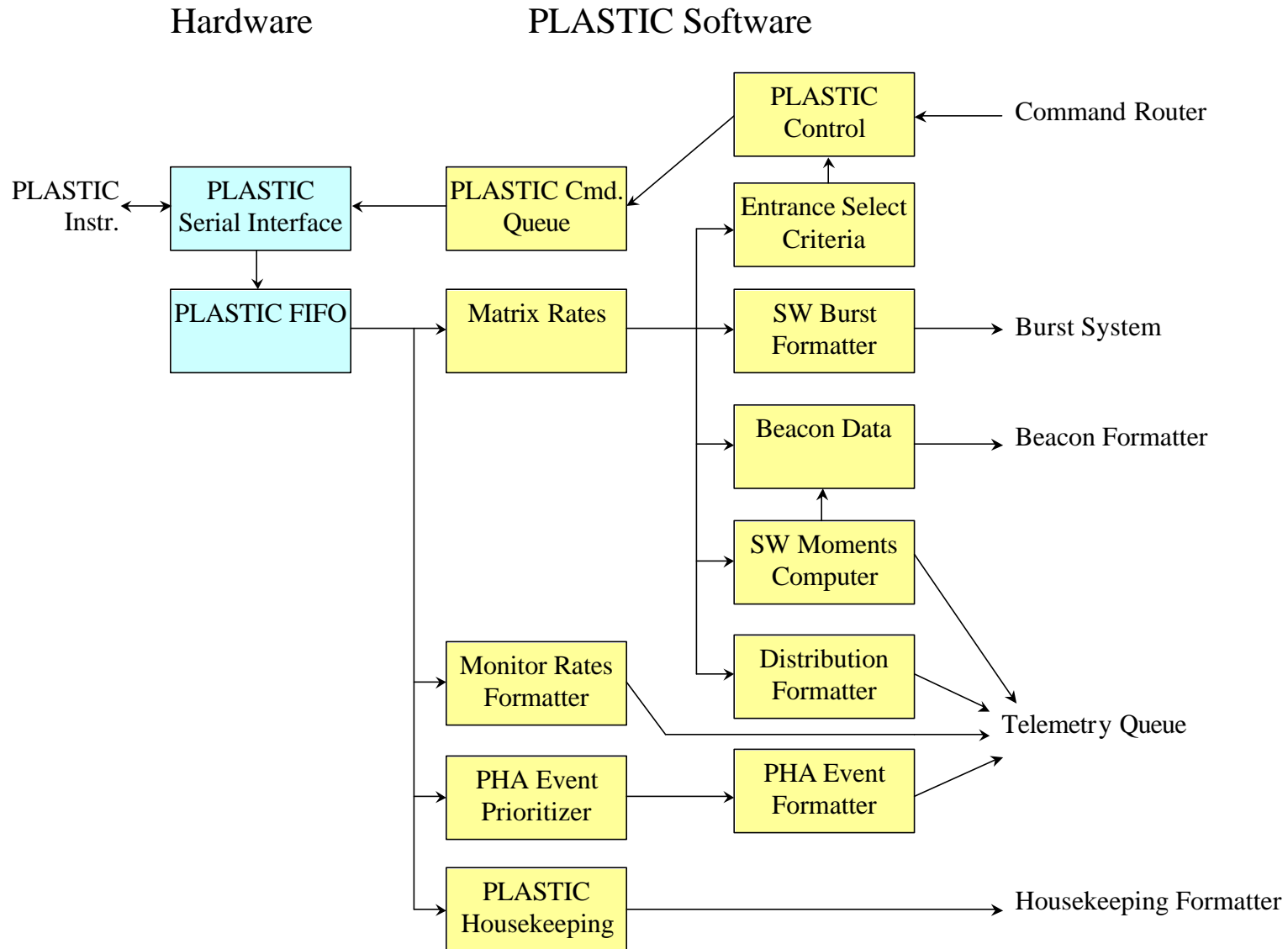












IDPU Flight Software Resource Estimates

Task	Processor Cycles, %	Code Size, kbytes	Buffer Memory, kbytes
Service Software:			
Operating System	1%	0.3 (b)	32.0
1553 Interface	1%	3.0 (b)	8.0
Command Router	<1%	0.7 (b)	10.0
Telemetry Packet Queue	<1%	0.3 (b)	10.0
Burst System	1%	1.0	1500.0 (*)
Serial Instrument Interfaces	1%	0.5	75.0
Housekeeping	<1%	0.2 (b)	0.6
Beacon Telemetry	<1%	0.2	0.6
Event Logging	<1%	0.3 (b)	0.6
Safing	<1%	0.5 (b)	0.3
MAG Software	1%	0.5	1.2
SEP Software	1%	1.0	2.5
SWEA Software:			
SWEA Moments	20%	1.0	17.0
SWEA Distributions	4%	2.5	10.8
SWEA Misc.	1%	1.0	2.5
STE Software	1%	1.0	3.7
PLASTIC Software:			
PLASTIC Distribution	6%	2.0	185.6
PLASTIC PHA	5%	0.5	46.6
PLASTIC Moments	5%	1.0	45.6
PLASTIC Misc.	1%	1.0	2.5
Total	55%	18.0	1955.1
Available	100%	64	3072
% Usage	55%	28%	64%

(*) Nominal value. Burst memory will expand to use all unallocated memory space. *Have recently added an extra Mbyte (2 chips) due to increased PLASTIC buffering requirements, and to improve Burst capability given greater telemetry bitrate availability*

(b) To be included (at least in part) in Boot PROM. PROM = 8kbytes, usage = 66%

Have recently modified memory decoding to allow 64Kbytes of code space