**Space Systems**

**Description**

Space is a challenging environment which requires special considerations in relation to robustness, reliability and fault tolerance of the hardware components onboard Spacecraft and the software and network systems which connect these hardware components together. The Space Systems module discusses some of the key challenges that Space presents and the areas of research which are currently taking place within the Space Technology Centre at the School of Computing which are of relevance to the world's major Space agencies.

**Planetary surface simulation**

Use of image processing techniques to generate realistic representations of planetary surfaces for testing Spacecraft and techniques, including planetary rovers.

**Entry,descent, and landing:**

The challenges of landing autonomous craft into environments such as planetary surfaces.

**Satellite data reception and processing:**

The Satellite Receiving Station in Dundee (www.sat.dundee.ac.uk) which forms part of the Space Technology Centre receives data from several USA satellites, and is responsible for archiving it, processing it an disseminating it to UK environmental scientists. Members of the Satellite Receiving Team will discuss these issues and there will be an extended visit to the receiving station where you will get to see the receiving station in action.

**SpaceWire:**

SpaceWire is an onboard data-handling network standard that is widely used by the World's space agencies (http://spacewire.esa.int). Dundee leads the development of SpaceWire and STAR-Dundee Ltd (http://www.star-dundee.com/) is a spin-out company from the Space Technology Centre which specialises in the development of test equipment for SpaceWire networks, including routers and network and data analysers. The Space Systems module will provide a detailed discussion of the SpaceWire standard and there will be an opportunity to use some of the devices which have been developed by STAR-Dundee Ltd, to further enhance your understanding of the topics discussed.

**Space Systems**

**Uses of Space**

What are spacecraft used for? - **Current Uses of space**

* Communications
* Navigation
* Weather monitoring
* Environmental monitoring
* Natural resources monitoring
* Disaster management
* Surveillance
* Near Earth Object Defence
* Science and Exploration

**Communications**

Largest sector of the space industry -Financially the most important application area

Some typical communications applications:

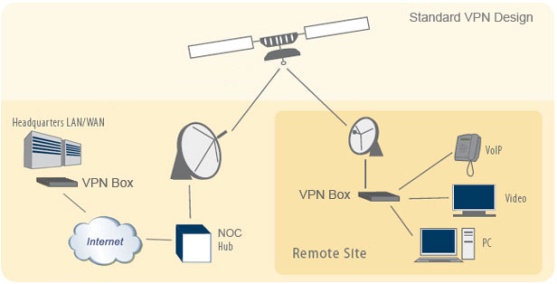
* Broadcast television and radio *e.g.* Sky
* Special television and radio services:
  + e.g. relays for live broadcasts etc.
* Satellite telephony:
  + mobile phone-type access for remote areas
* Broadband and internet relay:
  + especially for remote areas

**Broadcast Satellites**

* Single ground based transmitter
* signals received by satellite and retransmitted over wide area
* many ground-based receivers' can receive signal.

**Satellite must be at a fixed point in the sky**

- Broadcast satellites use Geostationary Earth Orbit(GEO)



**Global Navigation Satellite Systems**

Satellite-based positioning:

* An increasingly important space application

Two current systems operating:

* NAVSTAR GPS (USA)
* GLONASS (Russian)

Planned systems:

* Galileo (Europe) – completing in-orbit tests
* COMPASS (China) – test spacecraft launched
* IRNSS (India) – not global, good India coverage
* All based on similar technology

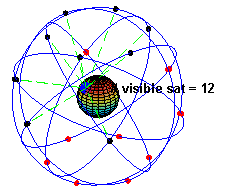
**GPS Constellation**

Only part of the constellation is visible:

* As the satellites move along their orbit around the Earth …
* … the maximum number an observer can see changes

GPS satellites use Medium Earth orbit (MEO):

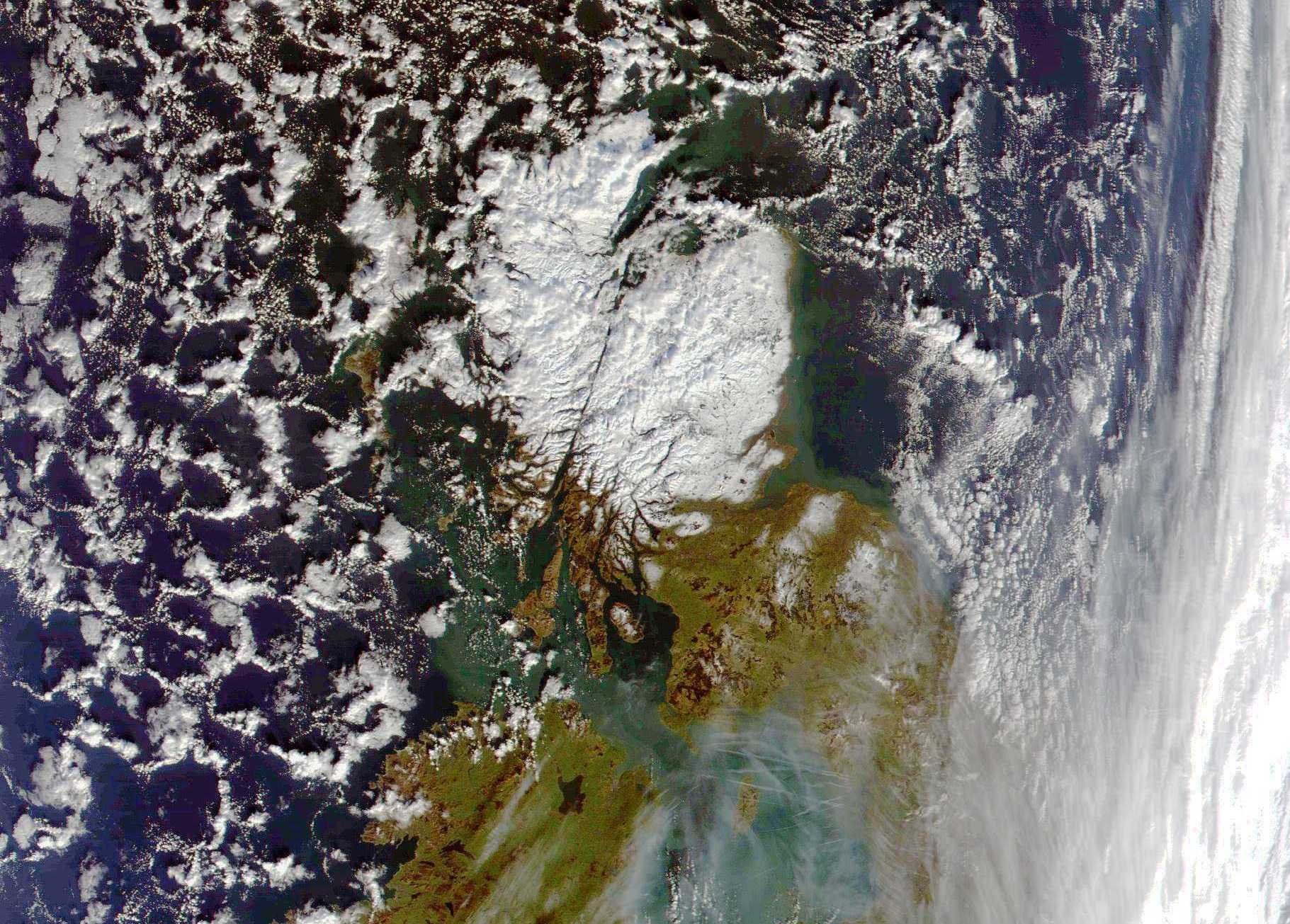
* + An altitude of 22,200km



**Meteorological Satellites**

Crucial for Short-term & long-term weather prediction

**Observe -** Clouds, Snow cover, Ice flows, Ocean currents, Fires, Sand and dust storms



**Disaster Monitoring**

Natural disasters:

* Require coordination of resources …
* … to target the needs of affected people

Satellites can play a crucial role:

* + Through observation and communication

Disaster Monitoring Constellation (DMC):

* + Provide observation of areas for disaster relief
  + Currently deployed and operated by:
    - Algeria, Nigeria, Turkey, UK and China
  + DMC satellites were designed in the UK
    - Surrey Satellite Technology Ltd



**What are the benefits?**

* Weather Prediction / Monitoring
* Disaster monitoring
* Communication

**Space Industry**

**Segments of the Space Industry**

**Launch segment:**

* + Getting spacecraft into space

**Space segment:**

* + Spacecraft

**Ground segment:**

* + Controlling the spacecraft from ground
  + Collecting data from the spacecraft

**User segment:**

* + Consumer (for satellite TV and GPS)
  + Scientists (for meteorological satellites etc)

**Launch Segment**

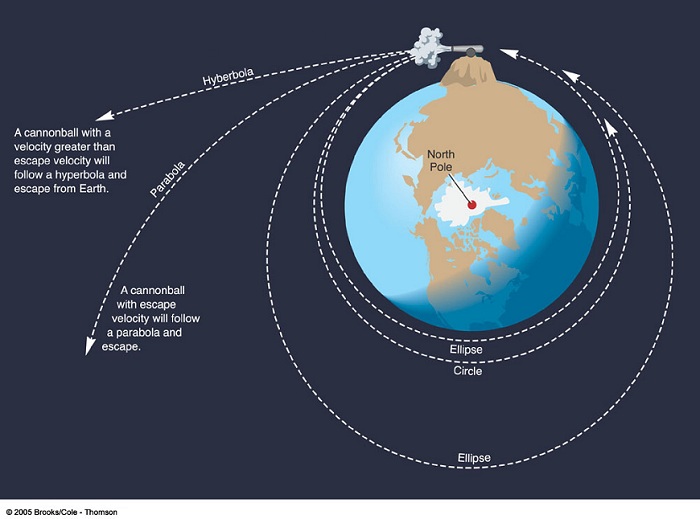
Responsible for:

* Placing the spacecraft (space segment) into space
* This may be into Earth orbit

**Getting Into space and staying there!**

Felix Baumgarter - Falling to earth

Ed white - floating in space



**Newton's cannonball**

In this experiment Newton visualizes a cannon on top of a very high mountain. If there were no forces of gravitation or air resistance, then the cannonball should follow a straight line away from [Earth](http://en.wikipedia.org/wiki/Earth), in the direction that it was fired. If a gravitational force acts on the cannon ball, it will follow a different path depending on its initial velocity. If the speed is low, it will simply fall back on Earth. (A and B) for example horizontal speed of 0 to 7000 m/s for Earth

Orbit - If the speed is the [orbital speed](http://en.wikipedia.org/wiki/Orbital_speed) at that altitude it will go on circling around the Earth along a fixed circular orbit just like the [moon](http://en.wikipedia.org/wiki/Moon). (C) for example horizontal speed of at approximately 7300 m/s for Earth

Elliptical Orbit - > higher than orbital velocity but lower than escape velocity.

Leave Earth - > higher than orbital + escape velocity.

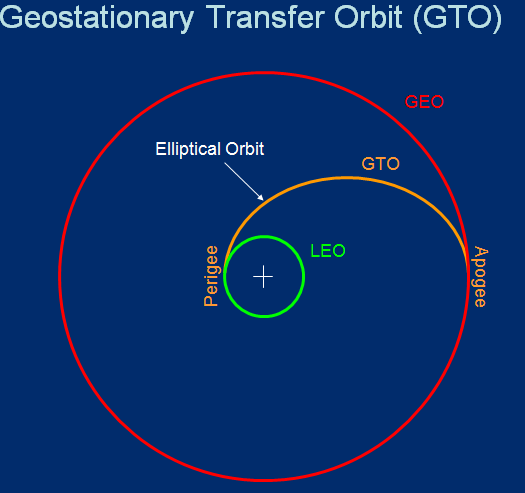
**Orbit Velocity depends on Orbit**

LEO - 7km/s

Orbit depencs on orbital velocity.

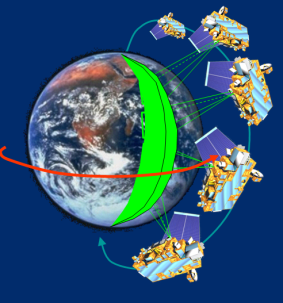
**Launching to Orbits**

* Launch vehicles deliver their payloads to LEO
* **To get to higher orbits:**
  + Start with an initial elliptical orbit (perigee in LEO)
  + Use thrusters/boosters on the spacecraft to change orbit
  + **Perigee = closest** to Earth, **apogee = furthest** from Earth
* **Getting to GEO:**
  + Payload placed in Geostationary Transfer Orbit (GTO)
  + GTO: perigee = LEO, apogee = GEO
  + When spacecraft reaches GEO apogee it ignites a rocket motor
  + This adds energy to circularise the orbit ...
  + ... leaving the spacecraft in GEO
* Similar idea for interplanetary missions



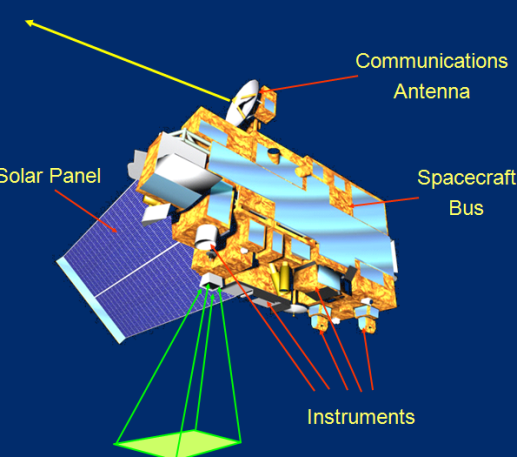
**LEO Pergee - > GTO Apogee -> GEO**

**How GEO satellites work**

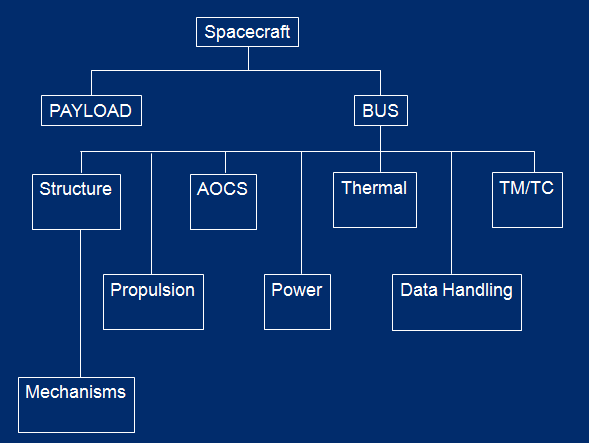


**Space Segment**

* Communications Antenna
* Spacecraft Bus
* Instruments
* solar panel



**Spacecraft system View**



**What factors make designing a spacecraft difficult?**

**Radiation**:

* + Affects materials and electronics

**Thermal**:

* + Extreme temperatures
  + Thermal management

**Vacuum**:

* + Outgassing releases contaminants
  + Changes to chemical/physical properties of materials

**Power**:

* + Not enough of it and generates heat

**G-forces and vibration during launch**

* + Careful mechanical design; vibration is hard to handle

**Mass restrictions**

* + Cost of launch
  + Lightweight materials and designs

**Micro-meteorites/debris**

* + Our atmosphere protects us but not the spacecraft

**Lack of maintenance:**

* + Can’t send a mechanic out at short notice!

**Spacecraft Payload**

**Overview of a Spacecraft Payload**

**Instruments and sensors**

* + Most applicable to science/EO missions

**Data processing**

* + May be computationally intensive

**Data handling:**

* + Mass memories
  + Data processors
  + Connection to TM/TC (telemetry/telecommand)

**Communications payloads:**

* + Receive and transmit data
  + Sometimes in very sophisticated ways

**Navigation payloads:**

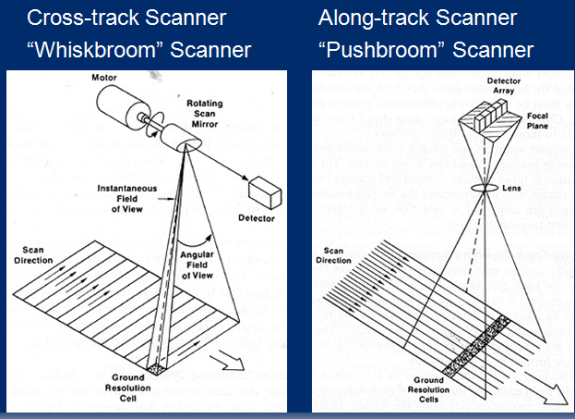
* + Transmit signals from an accurate clock

**Science Sensors**

**Scientific spacecraft carry a wide range of sensors**

* Optical (Cameras)
* Multi-spectral
* Hyper-spectral
* Radiometer
* Spectrometer
* Radar altimeter
* Ground penetrating radar
* Synthetic aperture radar
* LIDAR
* Magnetometer
* Gravitational/inertial sensing

**Sensor Scanning**

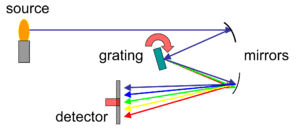
****

**Radiometers**

* Measure radiation
* Usually in a given band:
  + E.g. microwave
* Reflected and primary radiation
* Radiometers work with a broad band of frequencies/wavelengths
* Based on the technology of the sensor
* More precise bands can be obtained using a spectrometer

**Spectrometers**

* Examine electromagnetic spectrum in terms of frequency
* Uses a diffraction grating or prism to split light/waves into components, like a rainbow
  + Frequencies spread out spatially
* Brightness of each point indicates degree of frequency content
* Can be detected using a CCD



**Pushbroom Spectrometer**

* 2D CCD array
* X-dimension spatial (along “broom” edge)
* Y-dimension spectral (different frequencies)

**Common payload Services**

Onboard data-handling network

* + Sends commands to instruments
  + Collects data from instruments

Network used depends upon payload data rates

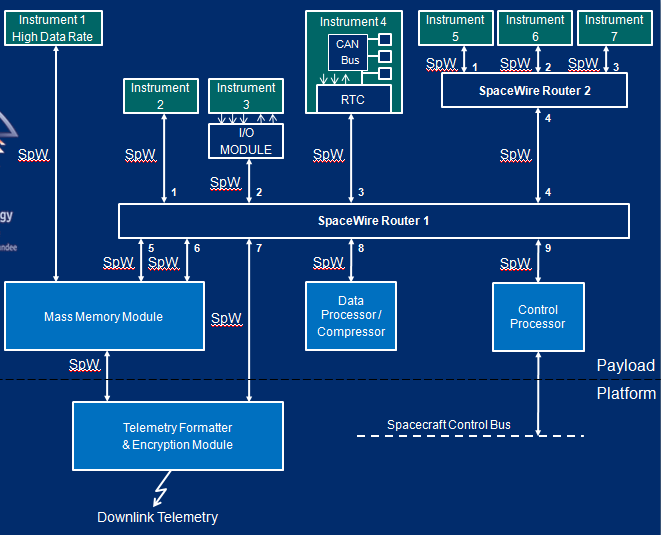
* + Optical 100’s Mbits/s
  + Radar and multispectral can be Gbits/s

SpaceWire

* + 200 Mbits/s

SpaceFibre

* + 2.5 Gbits/s



**Onboard mass memory**

* Used for temporary storage of data
  + Waiting to send it to ground
  + Can be compressed
* Solid state memory
  + 100’s Gbits to 10’s Tbits storage
* May be specifically for one payload
* May be shared among several instruments

**Solid State Mass Memory**

* Astrium GmbH
* Capacity: 1 to 1000 Gbit
* Mass: 10 Kg (384 Gbit)
* Interfaces:
  + IEEE 1355, SCSI, Parallel,
  + RS422, Mil STD 1553, ESA-OBDH,
  + QPSK-Telemetry
* Data rate: Input >4 x 200 Mbps, Output >2 x 150 Mbps
* Power: 40W ( 384 Gbit , data rate 800 Mbit)

**Instrument Processing**

* Processing normally instrument specific
* Occasionally a single processor will serve several instruments
* Radiation tolerant processors and memory
* Limited processing power and memory size
  + Compared to PCs
  + Makes software development challenging
* Real-time control

**Spacecraft Bus**

**Spacecraft Attitude and Positioning**

AOCS:

* Attitude and Orbit Control System
* Used for orbiting spacecraft
* Orbit raising
* Operational control

GNC:

* Guidance and Navigation Control
* Used for general manoeuvring
* Planetary landers
* In-orbit rendezvous

**Orbit Disturbances**

External:

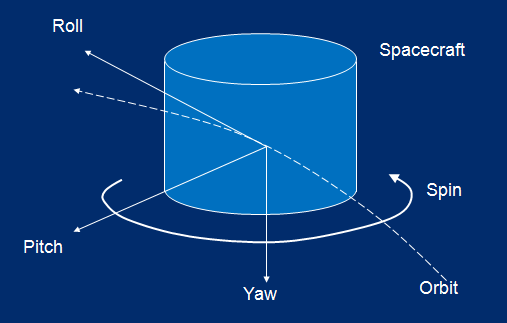
* + Solar radiation pressure
  + Magnetic field
  + Atmosphere and gravity gradient for LEO
  + Oblateness of the Earth (non-spherical shape)

Internal:

* + Thruster misalignments
  + Fuel slosh when manoeuvring
  + Operation of mechanical systems
  + Deployment of solar panels
  + Vibration modes, change in angular momentum

**Spin-Stabilised**

* Entire spacecraft spinning
* Gyroscopic effect keeps spacecraft stable



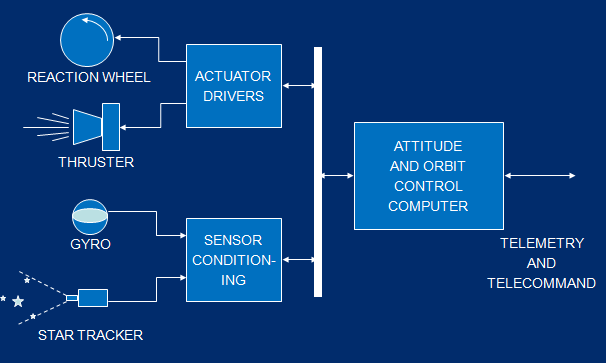
**Three-Axis Stabilised**

* Need spinning mass to keep spacecraft stable
* But do not want any part of the body of spacecraft to spin
* Use momentum wheels
* Masses spun up to provide angular momentum
* Electric motor can provide the spin
* What happens if you start them spinning in space?

**Propulsion Systems**

* Solid propellants
* Cold gas
* Monopropellant hydrazine
* Bi-propellant
  + Hydrogen and Oxygen
  + Another possibility:
    - Oxidant – Nitrogen Tetroxide N2O4
    - Fuel – Mono-methylhydrazine CH3NHNH2
    - Performance depends upon mixture ratio
    - 1.6 in this case
* Electric propulsion
* Solar sails

**Attitude and orbit control architecture**



**AOCS Sensors**

* Inertial sensors:
  + Determine changes in spacecraft
    - Position, Velocity, Acceleration
  + Uses a gravitational field (usually the Earth’s)
* Optical Sensors
  + Such as Sun, Earth and star sensors
* Magnetic sensors
  + Sense the Earth’s magnetic field
* Spacecraft in LEO can also use GPS

**Sun Sensors -** Analogue examples:

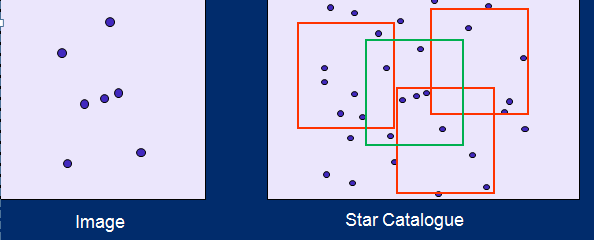
* Photocell output varies with Sun angle
* Bar throwing shadow over photocells
* Pairs of slits at different angles with photocells
* Digital:
* CCD or CMOS Active Pixel Sensor; image analysis to find Sun

**Star Trackers**

* Determine orientation relative to one or more stars
* Either known star
  + Track motion relative to the known star
* Or unknown star pattern
  + Take image of the stars
  + Determine which part of the star field it is

**Star Pattern Matching**

* Imaging sensor rigidly fixed to spacecraft body
* Wide field of view
* Take image of stars
* Match pattern to star catalogue



**Inertial Measurement Unit**

**Accelerometer:**

* Measures linear acceleration
* Integrate acceleration to estimate velocity/rotation
* Integrate velocity to estimate position/attitude
* Errors propagate so AOCS must compensate

**Gyroscope:**

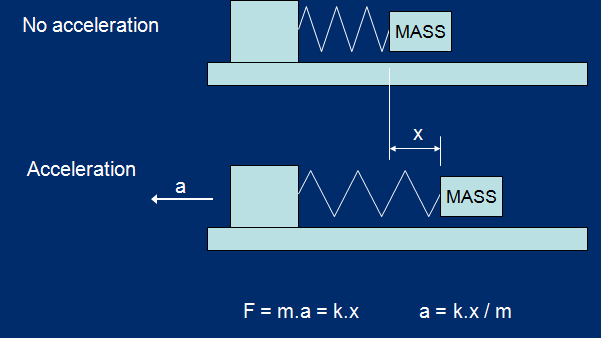
* Measures angular acceleration (rotation)

**Micro electro-mechanical system (MEMs):**

* These are becoming more feasible as technology improves

**Accelerometers**

* Sense acceleration
* Use a mass and spring system



**Gyroscopes**

* Measure angular acceleration or rate
* Principle of operation is either:
  + Gyroscopic effect
  + Coriolis effect
* Many different forms
  + Mechanical, based on spinning wheel
  + Micro-mechanical, based on a vibrating element
  + Optical, based on laser beams

**Attitude Control Actuators**

AOCS actuators:

* Drive the spacecraft into a desired attitude
* Compensate for disturbances
* Keep in required orbit

**Momentum and Reaction Wheels**

Allow spacecraft rotation about one axis

**Momentum wheels:**

* Large mass (up to 10kg), high speed (10000 rpm)
* Speed up or slow down to rotate spacecraft
* Need momentum dump if maximum speed reached
  + How can this be done?
* Use for stabilisation and large attitude changes

**Reaction wheels:**

* Small mass to be rotated in *either* direction
* Spin rates up to ±3500 rpm
* Reaction wheels are normally stationary …
* … or rotating slowly to avoid a jerky start
* Use for small changes in attitude

**Other Reaction Control Actuators**

**Magnetorquers**

* An electromagnet
* Interacts with the Earth’s magnetic field to cause spacecraft movement

**Reaction control thrusters**

* Very small rocket motors
* Picture shows principle
* Not a modern example

**Power Sub-System**

* Power provided by solar panels
* Batteries used to store charge
  + To provide power when in shadow of Earth
* Solar panels oriented towards Sun
  + To provide maximum power
* Power regulated
  + Distributed as 28 volt DC (typically)
  + To all electrical/electronic units

**Thermal Control**

* The Sun is very hot, Space is very cold
* Severe temperature differences
* Keep equipment within specified temp. limits
  + Operational temperature
  + Storage temperature
* Radiating panels
  + Pointing towards cold space remove heat
* Secondary surface mirrors
  + Reflect Sun’s energy
* Thermal blankets
  + Reflect Sun’s rays
  + Provide insulation

**Thermal Control**

* Thermistors used to measure temperature
  + At points all over spacecraft
  + Especially critical units
* Heaters used to control heat
  + But this takes power
* So try to use waste heat from some units to heat others

**Removing Heat**

* Often the problem is removing heat
  + E.g. use coolers for sensitive infra-red and visible instruments to reduce thermally induced noise in sensor
* Coolers remove heat
  + But it has to be disposed of
  + Coolers take substantial power
  + Use where very low temperatures required for payload sensor
* Good conductive path to radiators
* Radiators have to be shielded from Sun otherwise they will absorb the Suns’ energy

**Launchers**

**Spacecraft**

-Spacecraft Bus

-Spacecraft Payload

**Planet Surface Simulation**

* The descent of a planetary lander
* The Moon
* Simulating the lunar surface
* Evaluating the synthetic images
* Other planetary bodies

**Who or what has visited the Moon from Earth?**

Manned Landings

* + 6 Apollo missions (1969-1972)

Unmanned landings

* + USSR and USA had an unmanned lunar program in the 1950s -1970s
  + USSR first soft lunar landing in 1966
  + USSR also managed sample return missions in the early 1970s and a rover mission in 1973

Long gap!

* + Japan (Hiten1993), ESA (SMART-1), India (Chandrayaan-1 2008) and China (Chang’e 1 2009)

**Lunar Mission Successes**

**What was the first lunar lander: Who made it and when did it land? How did they do it?**

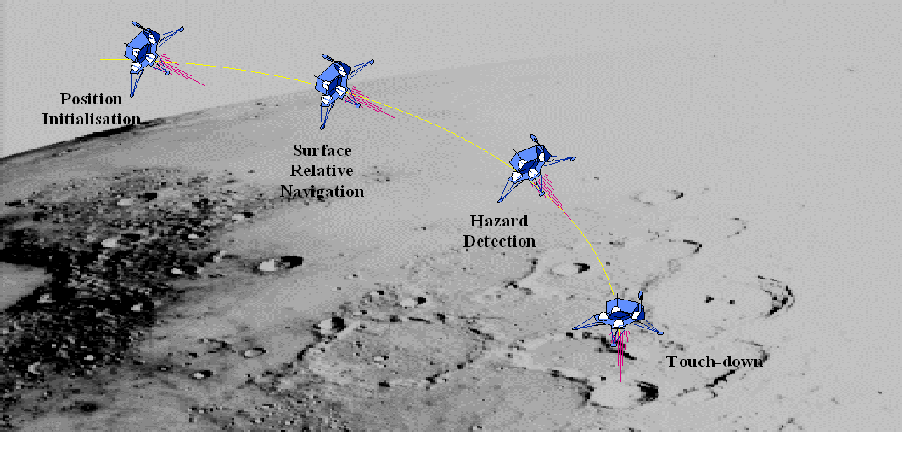
* 1959, Luna 9 - the first soft (robotic) lander ~15 m/sec!
* 1962, Ranger 4 – the first American (crash) lander
* 1966, Luna 9 - the first soft (robotic) lander ~15 m/sec!
* 1967 Surveyor 3, robot arm dug 17 cm trench
* 1970, Luna 16 sample return and Luna 17 had a rover which travelled >10km

**First Human Lunar Landing**

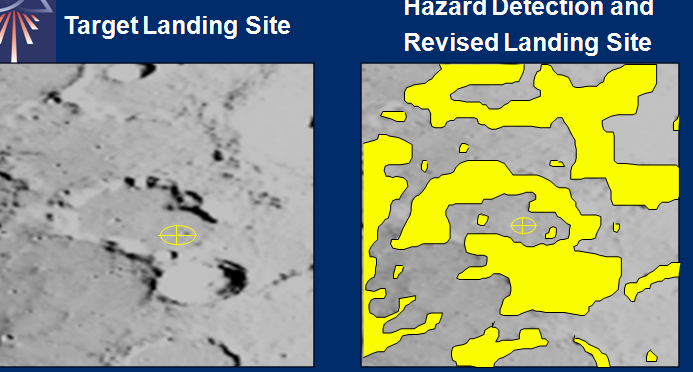
Apollo: 6 lander missions

* humans controlling a soft landing and returning

**Descent of an autonomous planetary lander**



**Hazard Detection**



**Testing Hazard Detection**

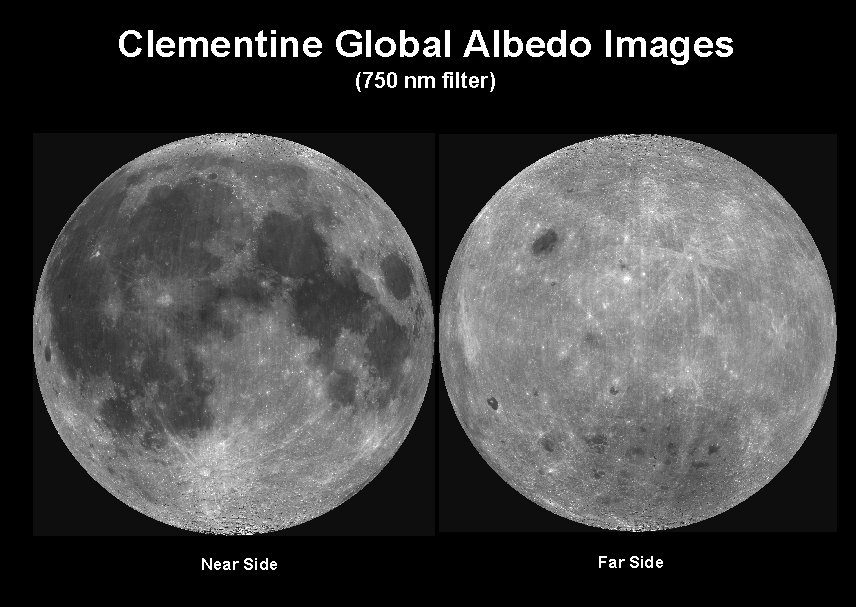
* Testbed
* Simuiliation
* Similiar enviroments

**virtual test environment of Hazard Detection**

* + More flexible
  + Cheaper
  + Easier to run series of tests
  + Better illumination simulation
  + More accurate calibration
* We started by looking at the Moon

**The Moon**

* Another planet hit the Earth
  + Massive collision
* Moon formed from debris
* No atmosphere
  + 14 orders of magnitude less than the Earth
  + Composed of the solar wind and lunar particles
* In fixed orbit around the earth
  + Same side faces the Earth at all times
  + Locked in tidal resonance
* Volcanic activity in the past
  + ceased around 3 billion years
* 16 km range of elevation



**What shapes the Lunar surface?**

What are the active geological processes?

* Volcanism
* Impact Catering

**Impact Cratering**

Formed by impacting meteorites

* hitting the surface at an average velocity of between 15-25 km/s

Mostly circular in shape

* Only around 5% craters are elliptical
* Why?

**Impact Process Stage 1**

**Contact and Compression**

* Transfers impact energy to the surface.
* Surface material is compressed at the point of impact, creating high-pressure shock waves.
* Material from both surface and meteor melts or vaporises due to the pressure and heat.

**Impact Process Stage 2**

**Evacuation**

* a hemispherical shock wave sets surface material in motion opening up a transient crater
* The diameter of the transient crater is many times larger than the projectile.
* Ejected material is dispersed radially forming an ejecta blanket.

**Impact Process Stage 3**

**Modification**:

* transient crater collapses, due to gravity,
* Final shape dependant on size and gravity.
* Uplift in the crater floor can lead to central peaks or peak rings in larger craters.



**Crater Types- Three categories of craters**

* Simple craters - Fresh Simple craters have sharp rims and bowl shaped interiors
* Complex craters - Extrusion in centre, like water droplet hitting surface.
* Multi-ring basins - Craters within craters

**Small scale surface features**

* Fractal terrain
* Simple craters
* Boulders

**Creating virtual lunar surface**

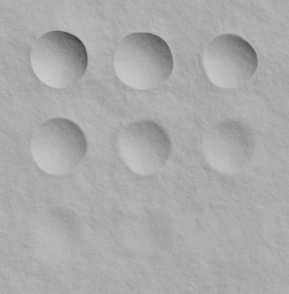
* Fractal terrain or initial real DEM
* Add craters
* Add boulders

**Crater Ageing**

* Craters are eroded by later impacts and settling dust over long periods of time.
* Effects of erosion:
  + Crater depth reduced
  + Crater rim height above the terrain reduced.
  + Crater rim becomes more rounded
* A severely eroded crater may appear as a depression with no rim.
* Eventually the crater may be completely obliterated.

**Eroded Crater Model**

* Erosional state of a crater
  + broadly dependent on its diameter
  + and the amount of cratering after it’s creation
* Crater depth and rim reduction
  + specified for a crater of a given diameter over time.
* Crater erosion for craters of other diameter can then be determined
* Craters are assigned an age when defined
  + from a user-definable age distribution.
* The Crater model is parameterised by
  + rim height
  + crater depth
  + normalised erosional state
* Baseline crater erosion over time defined
* Baseline crater erosion extrapolated to craters of other diameters



**Lunar South Pole Model**

* DEM data available at varying resolution
  + From Clementine images
  + From Earth based radar interferometry
* Can combine this data to create a base lunar south pole model
* It has holes in it! - These can be removed

**Evaluation**

* Visual comparisons
* Obvious artefacts
* Feedback from planetary scientists
* Image processing

****

**Modelling Mars from MOLA data**

* Can start with a low resolution real terrain model
* Then add in further terrain detail
* And extra features such as craters, boulders and dunes

**Advantages of Pangu**

* Landing a spacecraft on another planetary body is difficult
  + Testing essential but difficult
* Virtual testing is an important test facility
* PANGU has been created to model a variety solid planetary surfaces
  + Simulate camera and LIDAR
* Can enhance low resolution data sets to provide realistic lander test scenarios
* Flight path can be used to generate a set of test images with which to test lander navigation and hazard detection
* Simulate lander descent

**EDL**

**What is EDL?**

* Place spacecraft on the surface of a planet/asteroid/comet

**Why?!**

* Scientific experiments in situ
* Return samples to Earth for analysis
* Future: resupply existing infrastructure?

**How?**

* Different phases (atmospheric entry, deceleration, navigation,
* placement on surface) require different solutions.
* No direct interactive control from Earth

**EDL is very challenging**

* Most dangerous phase of mission
* Requires careful planning and clever onboard systems

**Wide range of scenarios**

* Moon/Mercury: no atmosphere
* Venus/Titan/Mars: atmospheres of different densities
* Asteroids/comets: very low gravity
* Hazards at all stages of EDL

**Each Step of EDL**

**Entry:**

– Place craft on desired trajectory

– Arrive in atmosphere from interplanetary space

– Various unknowns/perturbations:

• Thruster variation, sensor errors, environment knowledge

**Descent:**

– Stop from high speeds of e.g. 3-8 km/s (11000-29000 kph)

– Atmospheric effects: heating, braking, wind

– Navigation/flying to adjust for perturbations

– Maintain desired trajectory

– Landing site selection and hazard avoidance

– Must happen autonomously!

**Landing:**

– Deploying/activating landing system: legs/airbags

– Removing all residual velocity i.e. stopping

**Success of EDL in early Missions**

Most failed with crash landings, lost of contact, later missions became more successful.

* Redundancy was vital as early missions lost contact frequently.

**Computing Roles in EDL**

**Before Launch**

– Design and simulation of complete mission

– Selecting landing sites, time scales, requirements

– Planning/designing mission phases

– Analysing expected EDL performances

• Vital for new sensor types and advanced technology

**EDL**

– Complete onboard systems (hardware and software)

– Sensor data processing

– Fault detection, correction, recovery

**After Landing**

– Vehicle command and control; autonomous navigation

– Data gathering, transmission, analysis, etc.

**Landing Site Selection**

**Spacecraft are very complex machines**

– stuffed with computers of various kinds

– designed and built differently for each mission

where to land? ( Engineer, Scientific, Where it actually lands)

**Landing site considerations**

**Safety/Engineering demands:**

– Low winds (smooth descent through atmosphere)

– Smooth terrain (fewer landing hazards)

– Environment (temperature extremes etc)

– Polar regions bad (little Sun light)

– Dusty regions bad (sink into soil / obscure solar panels)

**Scientific demands:**

– Areas rich in rocks & layered terrain

– Areas with (water?) erosion features

Land sites balance safety requirements with scientific appeal

**Landing ellipse**

– Centred on the target landing site

– Indicate uncertainty in actual landing point

– Computed using geometry + landing simulations

– 280 100 km for Viking (1976)

– 200 100 km for Mars Pathfinder (1997)

– 80 12 km for Spirit/Opportunity (2004)

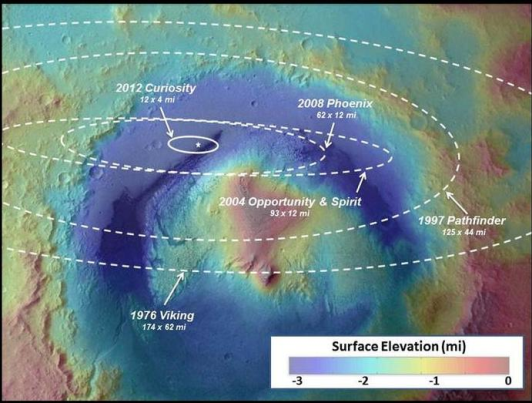
• Planned total distance around 1km each

• Spirit has travelled 7.7km (now dead)

• Opportunity has travelled 35km

– 20 20 km for Mars Science Laboratory (2012)

**Landings to date - Ellipses**



Improvements in technology make landing more precise

* **Inside craters are the most interesting areas.**
* **Less distance to travel the better**
* **Avoids landing in a huge crater.**

**Mars Rover - Case Study - EDL Phase**

* Aerobraking with small thrusters for guidance
* Parachute descent
* Powered descent with thrusters until hovering
* Sky crane to deploy rover to surface
* “Seven minutes of terror”

Entry Phase

**Strong gravitational field**

– Must slow down somehow

– Cannot return from surface: in-situ sample analysis

**Thick atmosphere**

– Must consider aerodynamic drag and other effects

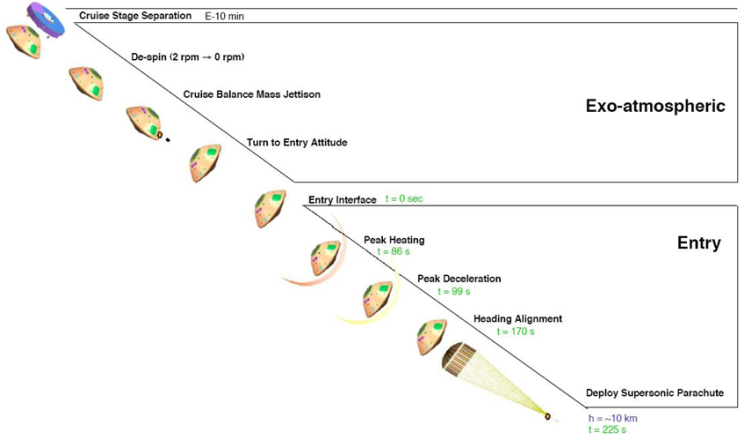
– Entry angle too shallow → miss or bounce

– Entry angle too steep → burn up or crash

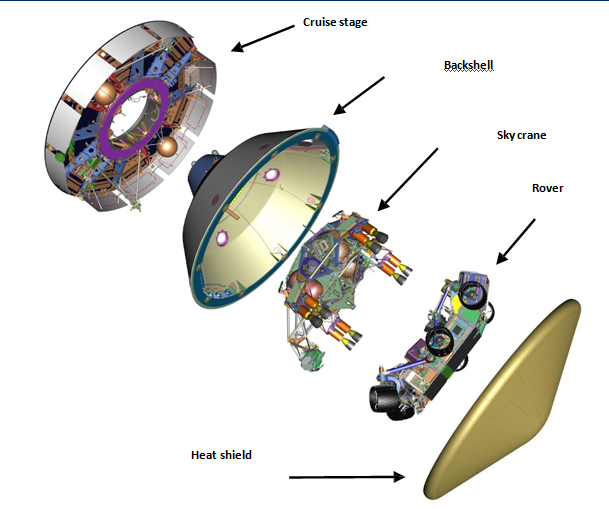
– NASA Mars missions used -11 to -17 entry angle

– 70 sphere cone aeroshell is used (from Viking missions)

– Use small thrusters to direct heading of craft



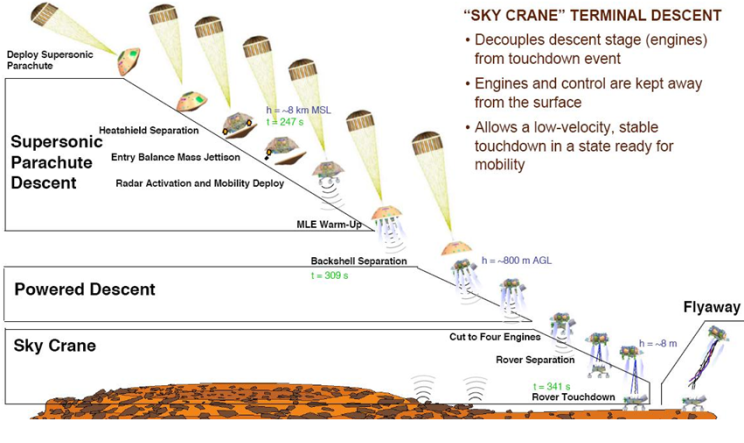
Cruise - > Drop - > Air resistance -> Parashoot - Sky Crane



**Descent Phase**

Supersonic parachute braking (225-309s, 10km)

Rocket powered descent (309-341s, 1.8km)



**Haya busa - Case Study - EDL Mission**

**Asteroid/comet link**

– What is the nature of comet nuclei?

**Bring back a sample of material**

– Unprocessed material from formation of the Solar system

– Composition of Solar nebula

– Planet formation models

– Timescales of significant events in the history of the Solar system

– Organic compounds?

**Advantages and Disadvantages of Asteroid landing**

**Advantages - Slower dynamics**

– More time for navigation decisions & algorithms

– Land at several centimetres per second

– Can descend to map surface, then retreat and plan landing

**Disadvantages** - Small, irregular gravitational field

– Makes navigation tricky

– Easy to miss completely!

– Sample collection trickier – no gravity to assist

**Summary of EDL**

**EDL is most challenging phase of mission**

Advancing technology - more precise landings

**Typical Mars EDL scenario:**

– Deceleration with heatshield and orientation with small thrusters

– Deceleration with parachute

– Deceleration with thrusters until hovering

– „Sky crane‟ deployment to surface

**Large planets and small bodies present different challenges**

**Space Wire**

**Introduction to SpaceWire Concepts and Principles**

* Review of space system requirements
* The onboard data handling problem
* Older data handling system solutions
* Introducing SpaceWire
* SpaceWire links
* SpaceWire networks
* SpaceWire routing

**Why Space Is Hard : Overview**

**G-forces and vibration during launch**

* Careful mechanical design; vibration is hard to handle

**Radiation:**

* Affects materials and electronics

**Temperatures:**

* Extremes of space and thermal management

**Vacuum:**

* Outgassing releases contaminants
* Changes to chemical/physical properties of materials

**Micro-meteorites/debris**

* Our atmosphere protects us but not the spacecraft

**Power:**

* Not enough of it and it generates heat

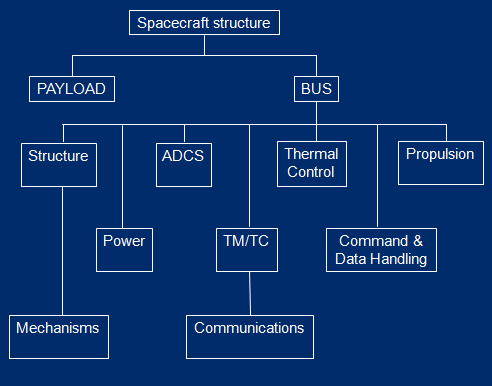
**Cost of design, build, launch & operation**

* orbit (LEO) From $5,000 to over $15,000 per Kg to low earth

**Lack of maintenance:**

* Can’t send an engineer to fix problems!

**Spacecraft System View**

****

**Typical Science Mission**

**Instruments – for example…**

* LIDAR Altimeter (low data rate)
* Infra-red radiometer (medium data rate)
* Imaging Camera (high data rate)

**Processing – for example…**

* Data Processor(s)
* Mass memory
* Communications processor
* Telemetry and telecommand (TM/TC) processor

**Communications**

* Instrument Data
* TM/TC

**The Problem**

**How do we get data between all of these devices?**

* Reliably
* Coping with radiation
* Coping with temperature extremes (and cycling)
* Coping with vibration
* Coping with a vacuum
* With appropriate bandwidth
* Without using too much power
* In a timely fashion

**This is the onboard data handling (OBDH) problem**

**Older OBDH Solutions**

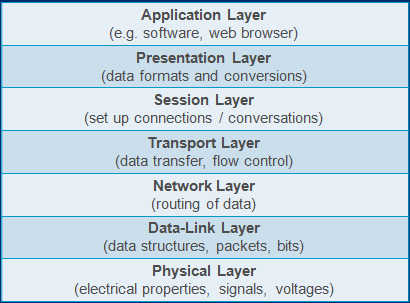
**All spacecraft have the OBDH problem -** Many solutions have been developed

**Examples**

* RS422/RS485
* MIL-STD-1553
* CAN
* USB
* SpaceWire

**Each of these has pros and cons -** Usually solved using a *data bus*

**OSI 7-Layer Model**

****

**Focus on Physical and Data-Link**

**RS422**

* Does not define anything beyond physical layer
* Only standardises electrical interface (not physical)
* Up to 10Mbps
* Higher voltage
  + E.g. +/- 6V compared to SpaceWire 350mV
* Point-to-point
* Differential signalling
* RS422 uses differential signalling over twisted pair
* Very reliable, even over long distances
* RS422 is *simplex*
* Data only goes one direction
* Use two connections for both directions: *full-duplex*
* Data is transmitted in short pieces, usually bytes
* Framed by start and stop bits to allow the receiver to synchronise

**Differential Signalling**

* **Digital signals are transmitted in binary**
  + Constructed out of ‘1’s and ‘0’s
* **Often, a high voltage indicates a ‘1’ and a low voltage indicates a ‘0’**
* **All voltages are referenced to ground**
  + This is called *single-ended signalling*

**Alternatively, transmit signal using two wires**

* + The difference between them indicates ‘1’ or ‘0’
  + One carries a positive voltage, the other a negative voltage
  + Swap over for ‘1’ or ‘0’
  + Always carrying opposites
  + This is called *differential signalling*

**Reliability and Twisted Pair**

* **Electrical signals are affected by *noise***
* Appears as a change in voltage level
  + Could turn a ‘0’ into a ‘1’ or vice-versa
* On a differentially encoded signal, the noise will normally affect both positive and negative portions *equally*
  + Therefore the *difference* will be the same and the signal will still be received as either a ‘1’ or a ‘0’
  + Noise that affects both parts of a differential signal is called *common-mode noise*
  + Noise that affects only one is called *differential-mode noise*
* To try and limit differential mode noise wires carrying +ve and –ve parts of signal are twisted together
  + Called *twisted pair cabling*

**Parity and Error Checking**

* Parity is a simple method for detecting single-bit errors
* Two types of parity: odd and even
* e.g. for odd parity:
  + Count the number of ‘1’s in the data byte
  + If it is odd, the parity bit should be a ‘0’
  + If it is even, the parity bit should be a ‘1’ (to make total odd)
* The receiver can then do the same, and check against the parity bit
  + If there is a discrepancy, an error occurred and the data cannot be trusted
* Used a great deal in networking

**RS485**

* Very similar to RS422
* Differential signalling
* Only defines physical layer
* Now permits transmission and reception on the same line
  + But not at the same time
  + This is called *half-duplex*
* Multiple units on the line
  + Shared medium
  + A bus
* Standard does not define how the medium is to be shared - This is called *bus contention*

**MIL-STD-1553**

* Developed by US military
* Very commonly used in aviation
* Shared differential twisted-pair bus (like RS485)
  + Half-duplex
* Standard defines physical, data-link and network layers
* Runs at 1Mbps
* Uses dual redundant buses for reliability
  + Called prime and redundant
  + Only one is used at a time, switches over if an error occurs
  + Uses odd parity for error detection
* Defined roles for devices on the bus
  + Bus controller (BC)
  + Remote terminal (RT)
  + Also bus monitor (optional)
* Uses command/response
  + i.e. ‘speak when you’re spoken to’
  + Solves bus contention problem
* Typical sequence:
  + BC addresses an RT
  + RT responds with its status
  + BC sends data to RT
  + RT responds with acknowledgement

**MIL-STD-1553 and Determinism**

* The time it takes for data to transfer from BC to RT (or vice versa) can be calculated at design time
  + MIL-STD-1553 is *deterministic*
* Communications across 1553 are orderly and data is always received in the order it was transmitted

**CAN (Controller Area Network)**

* Originally developed for, and most often used in, cars
  + E.g. Airbags, ABS, central locking, engine management etc.
* Shared differential twisted-pair bus
* Runs up to 1Mbps
* Defines physical and data-link layers
  + Standards for higher layers exist
* No bus master
* All bus *nodes* have a unique priority
* Electrical signals are encoded so that a node with a higher priority will always be able to detect bus contention, and will ‘win’
  + The ‘loser’ must *back-off* and *retry* later

**USB**

* Differential twisted-pair bus
* Up to 480Mbps (USB 2.0)
* Defines physical, data-link, network, transport layers
* Developed for personal computing
  + I.e. not ruggedised for space
  + Complicated software ‘stack’ required
* Higher power requirement (e.g. 5 volts).

**What About Ethernet...?**

* We haven’t looked at the most common network on Earth…
* Ethernet is attractive because protocol support for all 7 layers of the OSI model is well defined
  + Lots of existing software and know-how
* Is complicated and requires a lot of software
* Most of the complexity does not give features that are useful in space

**Problems with Older OBDH Solutions**

**RS422/485, MIL-STD-1553 and CAN are all**

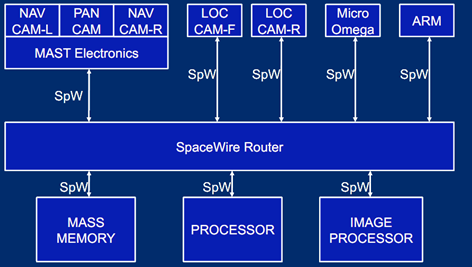
* Rugged (except USB and Ethernet)
* Fairly reliable
* Good against radiation, vibration, vacuum (in most cases)

**Problems**

* Low bandwidth (except USB)
* Lack of standardisation in the case of RS422/485
  + Propietary solutions
* Not full-duplex
* Higher power requirements
* 1553 Bus controller is a potential point of failure
* Architectural flexibility
  + Having to integrate with a restricted bus or restricted network topology instead of suiting missing specific requirements

**SpaceWire : An Overview**

* A network (not a bus, not a shared medium)
  + Built out of point-to-point links and routers
* Differential signalling over twisted pair
* Full-duplex links
* Up to 200Mbps
* Defines physical, data-link layers and part of the network layer
* Designed explicitly for space
* Very simple
* Low power
* Standard written by Prof. Steve Parkes here at Dundee
* **SpaceWire is being used all over the world, by all the major space organisations**



**SpaceWire Physical Layer**

* Cable
* Connectors
* How electrical voltages are used to convey basic information
  + i.e. binary 1 and 0

**SpaceWire Data-Strobe Encoding**

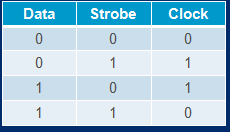
* Signals are encoded on two twisted pairs
  + Data
  + Strobe
* The strobe signal is an encoded version of a synchronisation signal called a *clock*
* The clock can be recovered very easily
* Data can then be decoded
* Downside: - Requires two sets of twisted pairs for each direction
  + Four twisted pairs in total

**Data-Strobe Encoding Details**

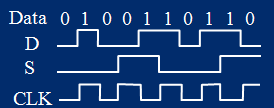
**Clock encoded and recovered using a simple logic function**

* *Exclusive-OR: true whenever inputs differ*

**Truth Table**

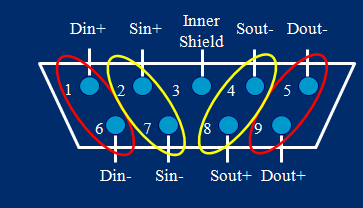
****

**Example**

****

SpaceWire Connectors

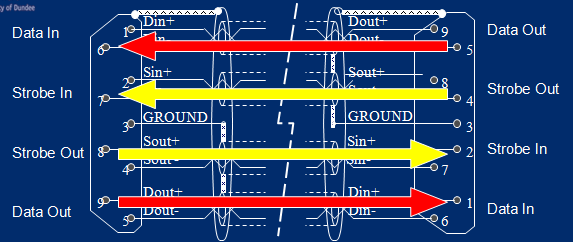
* 9-Pin Micro-miniature D-Type
* 8 signal wire arranged in four twisted pairs



SpaceWire Cable Assembly

Arranged so that transmit and receive signals cross over in the cable

* Transmit on one side becomes receive on the other
* Cable is symmetrical



SpaceWire Cable and Connector

* Rated for space
* Highly rugged
* Copes with temperature, vacuum, vibration etc.

PCB Tracks

* Can also be carried over *tracks* on *Printed Circuit Boards*
* Tracks routed in pairs, just like twisted pair

Spacewire Differential Signalling

* SpaceWire uses a standard called *Low Voltage Differential Signalling* (LVDS)
* Not susceptible to noise
* Does not generate noise
* Lots of existing hardware

**SpaceWire Data-Link Layer**

* **How information is transferred over the link**
* **Information is split into *characters***
* **These must be encoded and decoded**
  + Characters are converted to binary 0’s and 1’s
  + Binary 0’s and 1’s are converted to characters

**Characters on a SpaceWire Link**

**Two types of character on a SpaceWire link:**

* + Data characters (10 bits - P0xxxxxxxx)
  + Control characters (4 to 8 bits - P1xx)

**All characters protected by a parity bit**

**Control characters are special codes which indicate things about the link**

* + NULL characters (8 bits - P1110100)
  + Flow-control tokens (4 bits – P100)
  + End of packet markers (4 bits – P101)
  + Error end of packet markers (4 bits – P110)

**Null Charachters**

* A running SpaceWire link is *always* transmitting
* When there is no data to transmit, NULL characters are transmitted
* SpaceWire devices can detect as soon as there is a problem with a link
  + Even if there is no data being transferred
  + Detect loss of NULL characters

**Flow control**

* Each device on the end of a SpaceWire link has small buffers
* Called FIFO buffers
  + FIFO = First-In-First-Out
* Receiver may run out of buffer space
* Transmitter needs to know how much buffer space is left
* Receiver sends transmitter *flow-control tokens* (FCTs) which indicate it has space for 8 more data characters
* Transmitter keeps track of how much *credit* it has

**SpaceWire Packets**

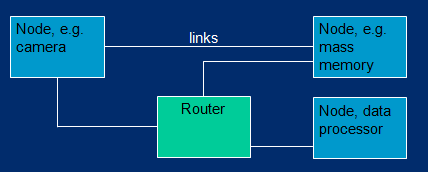
* Data is transferred in variable sized chunks called *packets*
* A packet can be anything from 1 byte to an infinite stream of bytes
* Each packet is made up of data characters and is finished with an *End of packet* (EOP) control character
* If an error occurred with transmitting the packet, and not all the data is transmitted, the EOP will be replaced by an EEP
  + An *Error End of Packet* control code

**SpaceWire Network Principles - Network Layer**

* **Connecting devices together**
* **How to get information to the right place - Routing**

**Spacewire Networks**

* **SpaceWire networks are made up of *links* connecting**
  + Nodes & Routers
* SpaceWire networks are ***packet switched***
* The alternative to this is ***circuit switching***

****

**Circuit vs. Packet Switching**

**Circuit switching is like making a telephone call**

* You wish to transfer some data (your conversation)
* You dial and the connection is made
* The connection uses network bandwidth even if you transfer no data (i.e. silence)
* The connection is only dropped when you put the phone down
* Routing by telephone exchanges

**Packet switching is like the postal service**

* Data is packaged into units
* Each unit is addressed
* Units all share the same medium and are interleaved
* Sorting offices do the routing

**SpaceWire Routing**

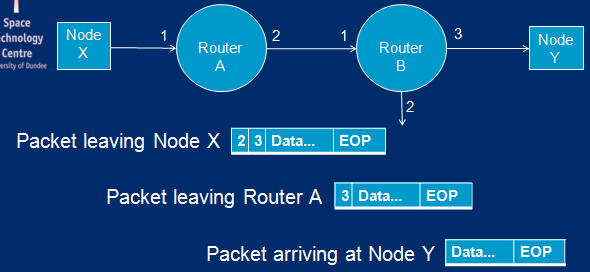
* SpaceWire is like the postal service
* The data characters at the beginning of each packet are treated as an address
* The rest of the packet is routed to where it needs to go by routers
* The router knows where the packet ends by finding the EOP character
* SpaceWire uses two forms of addressing
  + **Path addressing**
  + **Logical addressing**

**Path Addressing**

* Simplest form of addressing on SpaceWire is called *path addressing*
* Rather than specifying a destination, the path address specifies how to get there
* Each router has numbered ports (numbered from 1)
* The path address is simply a list of ports
* After each router routes, it removes the port number, exposing the next number for the next router

**Path Addressing Example**

* **Transfer a packet from Node X to Node Y**
* **Path address would be ‘2 3’**

****

**Wormhole Routing**

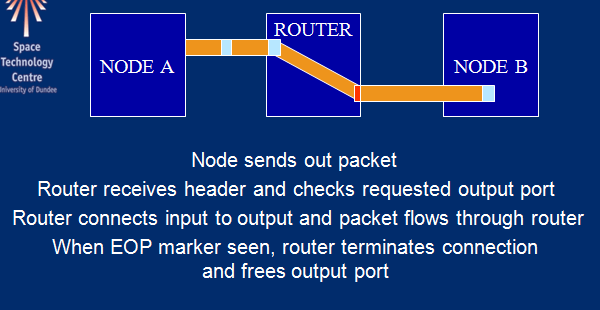
**Ethernet routers (for example)**

* Receive the whole packet
* Store it
* Decide where to send it
* Transmit it

**Called *store and forward***

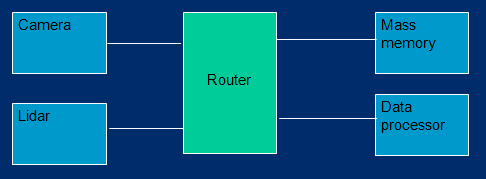
**SpaceWire routers don’t want to use all that memory (uses power, difficult in space)**

* Use *wormhole routing*
* Packet is not stored
* Router is transmitting the head of the packet even before it has finished receiving the tail

****

**SpaceWire Routers**

* SpaceWire routers use *crossbar switches*
* If the router is routing a packet from ports 1 to 2 it can simultaneously route a packet from 2 to 3
* All ports can be in use



**Routing Errors**

If the address at the head of a packet is invalid the router will discard the whole packet

* **Referred to a *spilling* the packet**

If a parity error occurs midway through a packet the router will spill the remainder of the packet an terminate the packet early with an EEP

* **If the error is at the head, the whole packet will be spilled**

If the output port is valid, but unavailable the router will wait

* After a timeout period, the router will spill the packet
* Unavailable could be due to a broken link or a busy link

**Logical Addressing**

SpaceWire routers can have up to 31 physical ports

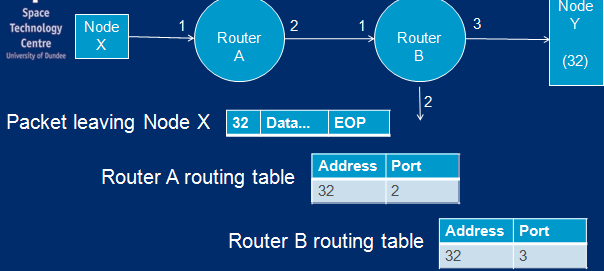
* **Port numbers 1 to 31**
* **Port number 0 is special (used for router configuration)**

**Addresses 32 and above (up to 255)** are treated differently

* These addresses are *logical addresses* indicating what the destination is rather than how to get there
* Each router recognises a logical address and looks up the number in an internal *routing table*
* The routing table contains an entry for each logical address and specifies the output port for that address
* Routers do not (in general) remove logical addresses after routing

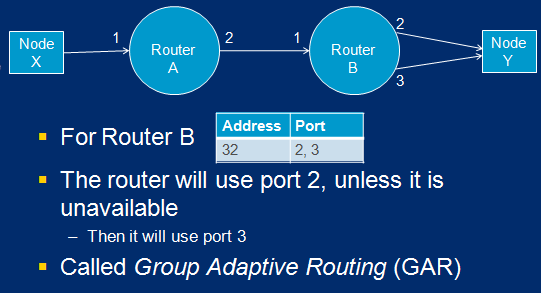
**Logical Addressing Example**

* Transfer a packet from Node X to Node Y
* Logical address would be ‘32’

****

**Group Adaptive Routing**

**Logical address routing can be extended by providing alternative ports in the routing table**

****

**Group Adaptive Routing – Why?**

**Higher Level Protocols**

* SpaceWire packets are designed to be simple, flexible, multi-purpose
  + You decide how to use it
* Common tasks on most missions:
  + Configure and control instruments
  + Read status information
  + Read / write data
* Performed in mission specific ways
* Opportunity for further standardisation
* Higher level protocols have been added to the SpaceWire standard to standardize these common tasks

**Defined SpaceWire Protocols**

* Not many defined SpaceWire protocols yet
* Some finished
  + **Remote Memory Access Protocol**
  + **CCSDS Packet Transfer Protocol (Space Packets)**

Some in the works

* + **SpaceWire-RT & SpaceWire-PnP**

**Protocol Identifier**

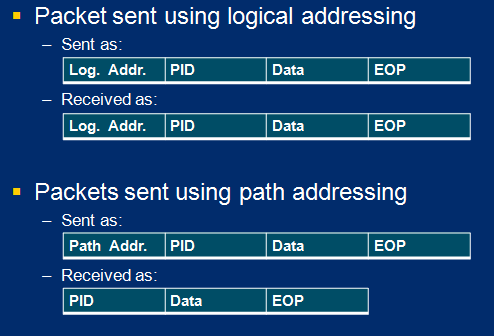
* Each protocol has a Protocol Identifier
* Uniquely identifies a protocol type
* Governed by the organisation that controls SpaceWire
* At its simplest: a single byte
* Placed at the front of a packet, straight after the address bytes

**Protocol ID Usage**

* Protocol ID 0 is reserved
* Protocol IDs 1-239 are controlled
* Protocol IDs 240-254 are available for general use (e.g. prototyping)
* Protocol ID 255 is reserved
* Two-byte, extended protocol IDs are also defined

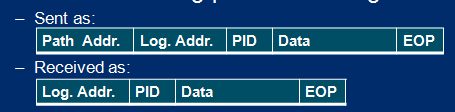
**Protocol Identifier Problem**

* A protocol ID is added to the head of each packet, after the address bytes
* If a packet is routed using logical addressing, the first byte of the packet when it reaches the node will be the logical address, followed by the protocol ID
* If a packet is routed using path addressing, there will be no address bytes left and the protocol ID will be first
* How can a node find the protocol ID?
* How can it tell the difference between an address and the protocol ID?

****

**Inserting a Logical Address**

* **To solve the problem we *always* insert a logical address**
* **Packets sent using logical addressing are unaffected**
* **Packets sent using path addressing**

****

* + **Now the node knows that the protocol ID is always the second byte**
* **The default logical address 254 is often used**

**Remote Memory Access Protocol**

* **A simple protocol to support two basic operations**
  + Obtain data from a remote source (read)
  + Send data to a remote source (write)
* **Implemented as reads and writes to memory addresses**
* The address could
  + **Refer to a memory location**
  + **Or identify a particular register**
  + **RMAP does not specify how the address and data should be interpreted**

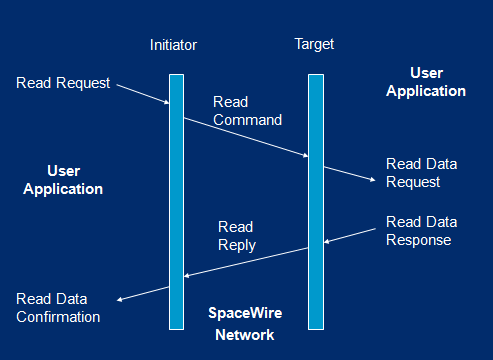
**RMAP Command Features**

* Either logical or path addressing can be used
* The protocol identifier is 1
* Each request has a transaction identifier
  + The reply will contain the same number
  + Allows an initiator to match a reply with an outstanding request
* A one-byte ‘destination key’ which must be set to a specific value for the command to be authorised
  + Value is defined by the implementer
* CRC used for error checking on both header and data

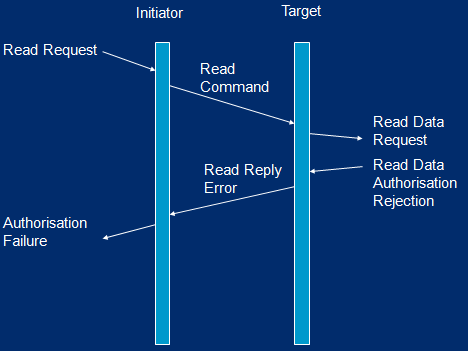
**RMAP Read Command**

* Requests a chunk of data
  + Of a specified length
  + From a particular location
* The ‘initiator’ sends a read request containing the address and the length
* The ‘target’ sends back a reply containing the data
* If the request was invalid for some reason, the reply may be empty and an error code will be specified
  + RMAP defines error codes

**Read Sequence Diagram**

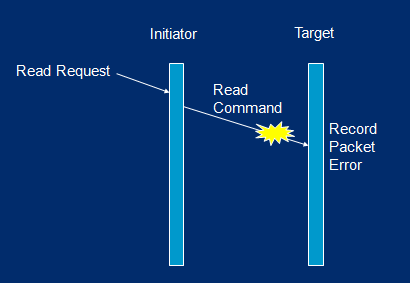
****

**Read Authorisation Rejection**



**Read Request Error**

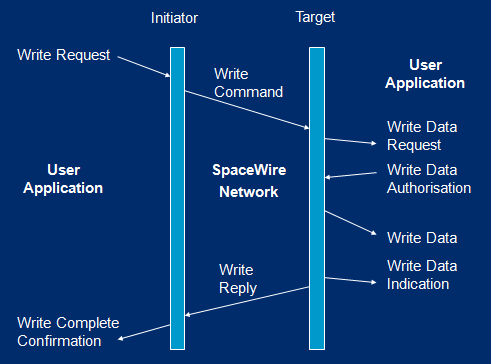
* The header CRC indicates an error



**RMAP Write Command**

* Various forms of write
* Simplest is where the initiator issues a write command with some data and the target does not reply
* Variations are based on
  + Acknowledgements: the initiator can request a reply after the write is completed
  + Verification: the initiator can specify that the target must check the data contents against the CRC before writing them

**RMAP Write Sequence Diagram**



**Verified and Acknowledged**

**Why not always verify?**

* To verify, the entire data cargo must be stored whilst the verification takes place and before the write happens (in case the CRC is bad). So this requires memory, time and complexity.

**Why not always acknowledge?**

* To acknowledge the target must be able to send a reply and (for it to be useful) the initiator must be able to accept it and process it. This just makes things more complicated.

**Read-Modify-Write**

* There is also a special read-modify-write command
* Packet from initiator to responder contains two fields:
  + Mask value
  + Data value
* Operation:
  + Responder reads the specified location
  + Data is operated on by the mask value
  + If some condition is met data is written (possibly using the mask value too)
  + The data which was first read is returned to the initiator
* What is this for?

**Uses of Read-Modify-Write**

* This is an atomic read-then-write operation
* Used for implementing locks such as semaphores
* Essential when two or more processes are accessing a shared resource
* Similar to multi-threaded applications

**Uses of RMAP**

* Basic reads/writes
* Reads/writes to a mailbox
  + A memory location acts as a ‘portal’ to a larger piece of memory
  + This is what non-incrementing addresses are for
* Event signalling
  + Carry out a read operation
  + Reply does not get dispatched until an event occurs
* Many more
* RMAP is simple but powerful and standardised!

**Viewing Earth From Space**

**Passive Sensors**

1. Sun’s radiation reflected from Earth
2. Radiation emitted by Earth

**Active Sensors**

1. Radiation artificially emitted by satellite, bounced off Earth’s surface and reflected back

Passive Sensors

* Mostly scanning imaging instruments and camera type devices

Atmospheric ‘sounders’

* + passive microwave system
  + observe noise (and hence heat/density) close to atomic resonance frequencies for height/temperature profiling
  + GPS used for atmosphere-grazing investigations
* Data rates low to very high, depending on the resolution.

Active Sensors

* RADAR altimeters for height measurement
  + Altimeter often provides wave height information as part of its operation.
* Synthetic Aperture Radar (SAR) for all-weather observation. Very high data rates and complex processing.
* Significant drain on onboard power, often only operational for 1/10 of an orbit and scheduled according to user priority system.
* Also LIDAR (covered elsewhere on course)

***From now on we will focus on passive imaging sensors - these are what we deal with at Dundee***

**“Multispectral” imaging**

* The radiation available & atmospheric windows are exploited.
* Can simultaneously observe different wavelengths producing multiple images
* Images commonly acquired using:
  + Visible Light
  + Near Infrared (“Near IR”)
  + Mid Infrared (“Mid IR”)
  + Thermal Infrared (“Thermal IR” or “TIR”)

**Why not use only visible light?**

* it can only show us so much
* clouds obscure surface
* no night images

**Thermal Infrared emitted by Earth surface is used to obtain images day & night**

* Not dependent on illumination by Sun
* Reveals temperature of Earth

Different wavebands are useful for different purposes such as:

* meteorology & water vapour in the atmosphere
* oceanography & marine biology
* vegetation, ecosystem and carbon studies

**Images from different wavebands can also be combined or ratioed to extract additional information**

* Possible to estimate physical properties of Earth’s surface
  + E.g. vegetation cover, biomass, etc
* Note however that these are estimates based on reflected radiation
  + Need “ground truth” data to validate

**Colour versus Monochrome**

* Monochrome image: each pixel is a single number: a brightness somewhere in the range from black to white.
* Colour image: each pixel is actually three numbers – a red, a green and a blue value
* So, an image from any single waveband will always appear monochrome
* To get a “true colour” image, satellite sensor needs to have separately measured & recorded:
  + Visible red
  + Visible green
  + Visible blue
* If the sensor just looks at the whole “visible” light spectrum at once the image will still appear monochrome as it’s just a single “brightness” value for each pixel

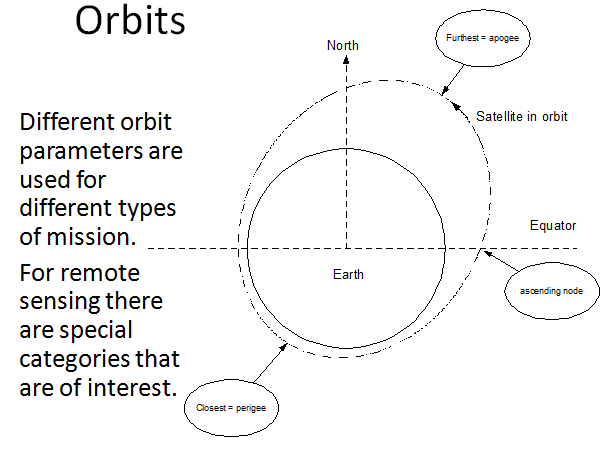
**The Remote Sensing Tradeoff**

* So, surely we want to image as many wavebands as possible?

**Visible red, Visible green, Visible blue, Near Infrared, Mid Infrared, ,Thermal Infrared**

* Problem: There is only a limited amount of radiation available for detection!
* Filtering out individual wavebands to produce loads of separate images reduces the radiation available for each
* To maintain an acceptable signal to noise ratio
* Also the higher the spatial resolution we want, the smaller the area on the ground each pixel represents
  + less available radiation to detect
* So, number and selection of wavebands a given satellite sensor will image depends on
  + Intended purpose (i.e. which wavebands are of interest)
  + Spatial resolution required
* Note that high spatial resolution often isn’t the top priority!

**Satellite Orbits**



Geostationary Orbit - near-circular orbit around the Earth’s equator at a radius of

Sun-Synchronous Orbit - Polar orbit

**Geostationary satellites**

**Advantages**:

* + *Continuous* monitoring of same area (high temporal resolution)
  + Simultaneous coverage of Earth disc
  + Simultaneous coverage of most of the Earth’s surface with just a few satellites
  + Low cost reception equipment (fixed, non-tracking antennas)

**Disadvantages:**

* + Satellites are much further away so spatial resolutions are relatively low
  + Spatial resolution varies significantly across the image due to curvature of earth across the large distances covered
  + No coverage of the poles (orbit is above the equator)

**Polar-orbiting satellites**

* Much closer to Earth but continually moving relative to Earth’s surface
  + typically only hundreds of km away rather than thousands!

**Advantages**:

* + More reflected radiation available for detection at this range
    - Higher spatial resolutions possible
    - More types of sensors can be used
  + Can observe the entire Earth’s surface – including polar regions

**Disadvantages**:

* + One satellite can’t continuously monitor the same area
    - Temporal resolutions are limited
  + Reception equipment is more complex & expensive
    - Tracking antennas required

**Sun-Synchronous Orbit**

* A type of **polar orbit** commonly used for remote sensing satellites.
* Result is a near-constant Satellite-Earth-Sun angle for good illumination of cloud structure, etc.
* Near circular orbit with correct combination of period and inclination so that orbital plane precesses by 360° per year due to the Earth’s equatorial bulge (J2 gravity term).
* Typical combination is around 102 minutes (~7230km semi-major axis so ~850km altitude) and 98.8° inclination.

**Space Systems - Revision - Class Test 2010**

Q1 **: The Attitude and Orbital Control System (AOCS) of a spacecraft uses sensors and actuators in order to maintain a spacecraft in the correct orbit and orientation. Briefly describe two types of sensor, and two types of actuator commonly used in AOCS**

**Two Types of Sensors**

**Relative attitude sensors**

* Gyroscopes
* Motion Reference Units
  + Antenna motion compensation and stabilization
  + Dynamic positioning
  + Heave compensation of offshore cranes
  + High speed craft motion control and damping systems
  + Hydro acous
  + tic positioning
  + Motion compensation of single and multibeam echosounders
  + Ocean wave measurements
  + Offshore structure motion monitoring

**Absolute attitude sensors**

* Horizon sensor
* Orbital gyrocompass
* Sun sensor
* Earth sensor
* Star tracker
* Magnetometer

**Types of Actuators** - anything to do with the control of movement of the spaceship

* Thrusters
* Monumentum Wheels / Reaction wheels
* Spin stabilisation
* Gravity gradients

========================================================

Q2 . **In terms of spacecraft systems, what is meant by the term ‘bus / payload split**

**Bus -** the infrastructure of a spacecraft, usually providing locations for the payload - The bus or payload platform consists of the subsystems that support the payload

**A bus typically consists of the following subsystems:**

* Command and Data Handling (C&DH) System
* Communications system and antennas
* Electrical Power System (EPS)
* Propulsion
* Thermal control
* Attitude Control System (ACS)
* Guidance, Navigation and Control (GNC) System
* Structures and trusses
* Life support (for crewed missions).

**Payload -** Scientific Instruments

========================================================

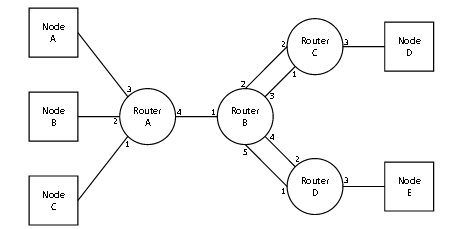
Q3 .In the following network diagram links are represented by lines, and numbers

give the port number to which that link is connected.

i. If a packet is to be routed from Node A to Node D, give a path address that should be included at the head of the packet.

ii. If a logical address of 32 is to be used to route a similar packet from

Node A to Node D, give the relevant routing table entry for Router B, making use of Group Adaptive Routing



i - Path Address :

4 2 3    or    4 3 3

ii - Logical Address :

Node A : 36

Node B : 35

Node C : 34

Node D : 32

Node E : 33

**ROUTER B** : Routing Table : :

HOST 36     PORTS   1

HOST 35     PORTS   1

HOST 34     PORTS   1

HOST 33     PORTS   4,5

HOST 32     PORTS   2,3

From ROUTER B to HOST D USE PORT 2 or 3

========================================================

**Q5. A spacecraft includes a Mass Memory Unit which stores information from three Instruments.Each Instrument transmits data over SpaceWire to the Mass Memory Unit using the Remote Memory Access Protocol (RMAP). A Processor Unit also uses RMAP over SpaceWire to read the instruments’ data from the Mass Memory Unit.**

i.Explain the advantages of using RMAP to access the Mass Memory Unit.

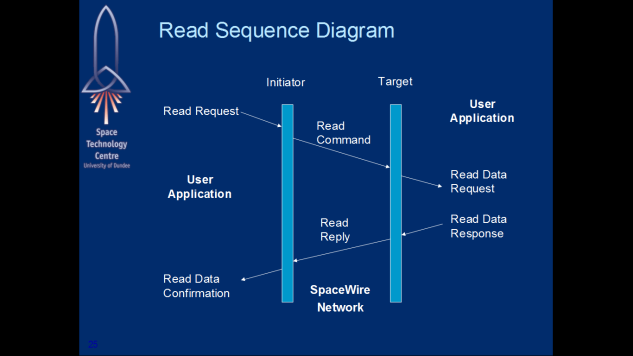
**RMAP is like DMA ( Direct Memory Access ).**

* Allows multiple devices to access the same Mass Memory Device but not the same blocks.
* Saves on processing power
* Allows low data rate instruments to share the same memory.
* Saves on Payload mass
* RMAP protocol is a standard so allows different devices to share the same memory.

**The Processor performs RMAP read commands to read Instrument data from the Mass Memory Unit. During one read operation, an error on a link causes the Mass Memory Unit’s response to be lost.**

**ii.Using a sequence diagram, describe the sequence of events which occur when this operation is performed and subsequently fails.**

RMAP Preformed and Passes.



Instead of Read Data response, Nothing is returned as the address and data cannot be verified.

iii.Describe how this error could be dealt with by the Processor, and the mechanisms that could be included to reduce the impact of errors.

Rerun on timeout waiting for response

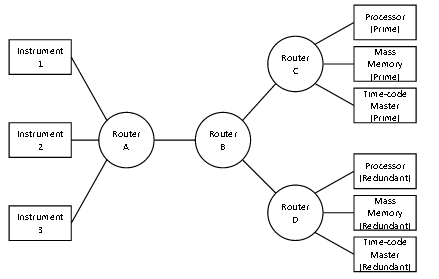
========================================================

**Q6 . Engineers are testing the SpaceWire network shown below. The prime and redundant Time code Masters are set to send time - codes once every second when enabled, and must send their first time code within 10 milliseconds of being enabled.**

**During testing it is found that:**

* **Both Time code Masters behave as expected when the other is powered down.**
* **When the prime Master is powered up 3 seconds after the redundant Master, the prime Processor and Mass Memory receive no time codes.**

**Describe the sequence of events which results in time codes never reaching the prime Processor and Mass Memory**



?????

========================================================

**Q7. Recently there has been considerable interest in sending space probes to theSouth Pole of the moon. For example the proposed ESA Moon - NEXT mission. Explain what makes this region such an interesting target.**

* As the south pole is never exposed to the sun any potential water trapped in the surface of the moon will not have evaporated.
* Significant amount of water found in a crater
* Water provides a potential resource needed to produce a moon base
* It can also be used to produce hydrogen fuel that would be used as a launch platform to marks.
* Lower gravity makes launches easier.

========================================================

**g. A lunar South Pole lander will be required to navigate with great precision and to be able to avoid hazards such as boulders. Providing realistic test data for the guidance and hazard - avoidance system of such a lander is made difficult by our incomplete  knowledge of the topography of this part of the moon.**

**Describe how the PANGU software can be used to fill in the gaps in our knowledge and simulate landing hazards realistically.**

* Only low resultion images are available of landings on the moon, as debris
* Landings have to be automated due to the time delay between
* Use what is known to build a model and extrapolate to fill the gaps
* Generation of fractal terrain, craters and boulders
* Lots of different surfaces can be modelled to test various possibilities

========================================================

**Class Test 2009**

**1.The environments experienced by spacecraft during launch and operation present significant challenges to spacecraft engineers. Briefly describe four of these challenges**

* Radiation - affects materials and electronics
* Power - not enough of it and generates heat
* Mass restriction - Cost of launch, lightweight material and design.
* Lack of maintenance - Cant send someone to fix something, redundancy.

2. Most communications satellites orbit the Earth around the equator and at analtitude of 36,000km. What is this orbit called, and why is it used in this way?

Types of orbits

1. [Polar - See every part of the earth as the earth rotates.](http://marine.rutgers.edu/cool/education/class/paul/orbits2.html#1)
2. [Sun Synchronous - 700-800km - Pass over at same time daily](http://marine.rutgers.edu/cool/education/class/paul/orbits2.html#2)
3. [Geosynchronous - Geostationary orbits, 36,000km](http://marine.rutgers.edu/cool/education/class/paul/orbits2.html#3)

Geostationationy orbit, Allows a single satellite to rotate around the each every 24 hours in the same position e.g. Sky TV

3. Compare and constract MOC2DIMES and PANGU

MOC2DIMES => Only using the given data, no attempt to extrapolate

PANGU => Generation of planetary surface combined with given data

3b. Describe the relative advantages of software test beds and mars anaglogue enviroments for testing vision based guidance system for mars landers.

**4. i  Path address for node a to Node d**

3 4 1  or 3 3 1

**logical address using Group Adaptive Routing**

Node a   32, Node b  33, node c   34, node d   35

Routing table

HOST 32 PATH 1

HOST 33 PATH 1

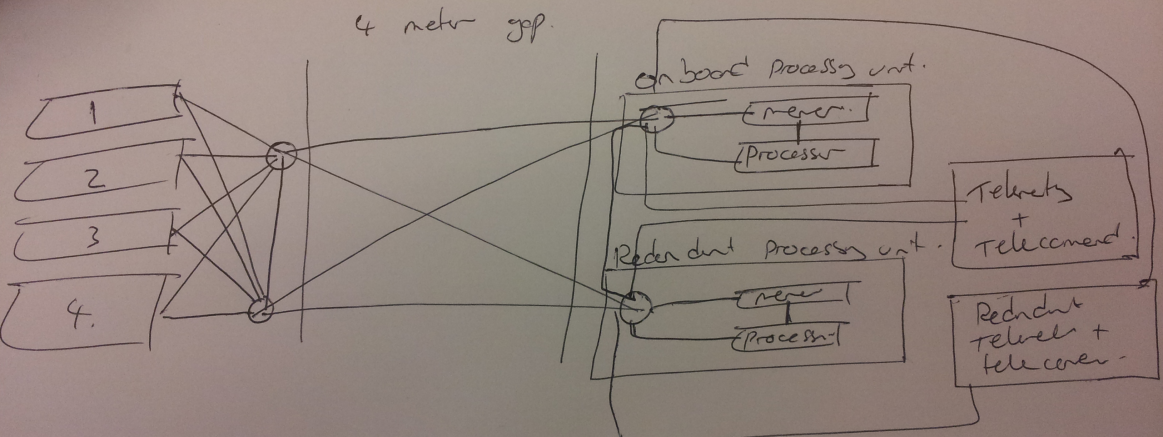
HOST 34 PATH 3,4

HOST 35 PATH 3,4

**4 b)**

* Multiple single points of failure, every instrument has no redundancy
* there is no redundancy for the router.
* No redundancy for main memory or process,
* single point of failure for telemetry and telecommand
* Multiple cables spanning over 4 meters across the bus.

If any component, instrument, or cable fails they are no longer accessible across the network.



**4 c ) Default data rate is slower than the maximum, so it builds up to it instead of large bursts of data.**

**EDL**

d = 2 \* h \* tan(f/2)                d = 2 \* 2000 \* tan(15)

t = (frames -1) \* time between frames    t = (50-1) \* 0.3

s = d / t                     s = 1071 / 14.7

meters per pixel =

    mpp = d / resolution            mpp = 1071 / 512

Trajectory =

    mpp / frames                Trajectory = 2.04530771875 \* 35

X Trajectory

Speed = Trajectory

X        35(pixles) = 2.04530771875\*35 = 71.58577015625 Meters in the X Direction

       Speed = 71.58577015625 / 14.7 = 4.8697802827380952380952380952381

========================================================

Variables

========================================================

Height = 2000m

Field of view = 30 degrees

Resolution = 512 pixels

=======================================================

Estimate length of descent sequence

========================================================

For any given number of frames, subtract one, then multiple by field of view in degrees, and this will give the descent sequence in seconds.

Given 50 frames, the length of the descent sequence is (50-1) x 0.3 = 14.7

   (50 Frames - 1) x (0.30) degrees field of view = (14.7) Descent Sequence

========================================================

Estimate horizontal velocity of the spacecraft.

========================================================

Constant height of 2000m above the surface

Identify 1 | 2 good tracks ( long and no zig zag)

Calculate x & y coordinates of spacecraft velocity.

========================================================

Find the meters

========================================================

Calculating the radians

   Tan(15) = 0.261799388 radians

Find the value of D

   d = 2 \* h \* tan(f/2)

   d = 2 \* 2000 \* 0.261799388 = 1047.197552

Divide D / 2 to find total distance in meters for triangle

   d/2 = 523.598776

Find the value of a pixel by diving distance / (resolution/2)

   523.598776/256 = 2.04530771875

Result

   each pixel is 2.04 meters

===========================================

Calculating the speed - Trajectory 1

===========================================

Speed = distance / time

X        35(pixles) = 2.04530771875\*35 = 71.58577015625 Meters in the X Direction

       Speed = 71.58577015625 / 14.7 = 4.8697802827380952380952380952381 seconds

Y       60(pixles) = 2.04530771875\*60 = 122.718463125 Meters in the Y Direction

       Speed = 122.718463125 / 14.7 = 8.34819477041 seconds

===========================================

Calculating the speed - Trajectory 2

===========================================

Speed = distance / time

X        52(pixles) = 2.04530771875\*52 = 106.356001375

       Speed = 106.356001375 / 14.7 = 7.2351021343537414965986394557823

Y       91(pixles) = 2.04530771875\*91 = 122.718463125

       Speed = 186.12300240625 / 14.7 = 12.661428735119047619047619047619