Astronomical images

Standard image format in astronomy

- Flexible Image Transport System (FITS)
- transporting, analyzing, and archiving scientific data
- much more than just another image format (e.g. JPG, GIF)
- store scientific data sets consisting of **multidimensional** arrays (images)

Always evolving...

...Looking for a standard format

- 1979: Initial use and interchange of FITS files
- 1981: Published original (single HDU) definition paper
- 1982: FITS format is formally endorsed by the IAU
- 1988: Defined rules for multiple HDUs in a FITS file
- 1988: FITS Working Group established by IAU (IAUFWG)
- 1988: FITS definition extended to include ASCII TABLE extensions
- 1990: FITS definition extended to IEEE floating-point data
- 1994: FITS definition extended to multiple image arrays in IMAGE extensions
- 1997: Adopted a Y2K-compliant date format
- 2001: Reiterated existing standard in one paper (NOST doc.)
- 2002: Approved conventions for generalized world coordinates and celestial coordintates.

Structure of a FITS

- Segments called **Header/Data Units** (HDUs)
- first HDU is called the *Primary HDU*, or Primary Array
- Primary data array can contain a 1-999 dimensional array of 1, 2 or 4 byte integers or 4 or 8 byte floating point numbers.
- A typical primary array could contain a 1-D spectrum, a 2-D image, or a 3-D data cube
- Any number of additional HDUs may follow the primary array. These additional HDUs are referred to as FITS **extensions**

Header Units

- Every HDU consists of an ASCII formatted **Header Unit** followed by an optional **Data Unit**.
- Each header or data unit is a multiple of 2880 bytes long.
- Each header unit contains a sequence of fixed-length 80-character **keyword records** which have the general form:

KEYNAME = value / comment string

Keyword names

- In standard FITS, the keyword names may be up to **8 char.** long and can only contain uppercase letters A to Z, the digits 0 to 9 and the underscore character.
- The keyword name is (usually) followed by an **equals sign** and a **space character** in columns 9 and 10 of the record,
- followed by the **value** of the keyword which may be either an integer, a floating point number, a complex value (i.e., a pair of numbers), a character string (enclosed in single quotes), or a Boolean value (the letter T or F).
- Some keywords, (e.g., COMMENT and HISTORY) are not followed by an equals sign and in that case columns 9 - 80 of the record may contain any string of ASCII text.

Mandatory Keywords

Each header unit begins with a series of **required keywords** that specify the size and format of the following data unit.
A 2-dimensional image primary array header, for example, begins with the following keywords:

```
SIMPLE  =                    T / file conforms to FITS standard
BITPIX  =                   16 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                  440 / length of data axis 1
NAXIS2  =                  300 / length of data axis 2
```

Other Keywords

- The required keywords may be followed by other **optional keywords** to describe various aspects of the data, such as the date and time of the observation, the exposure time, all informations about pointing, coordinates (RA,DEC) grid, etc..
- COMMENT or HISTORY keywords are also frequently added to further document thecontents of the data file
- The **last keyword** in the header is always the **END** keyword which has blank value and comment fields.
- The header is padded with additional blank records if necessary so that it is a multiple of 2880 bytes (equivalent to 36 80-byte keywords) long.

All of this was the “old” standard

Data Units

The data unit, if present, immediately follows the last 2880-byte block in the header unit. Note that the data unit **is not required**, so some HDUs only contain the header unit.

The image pixels in a primary array or an image extension may have one of 5 supported data types:

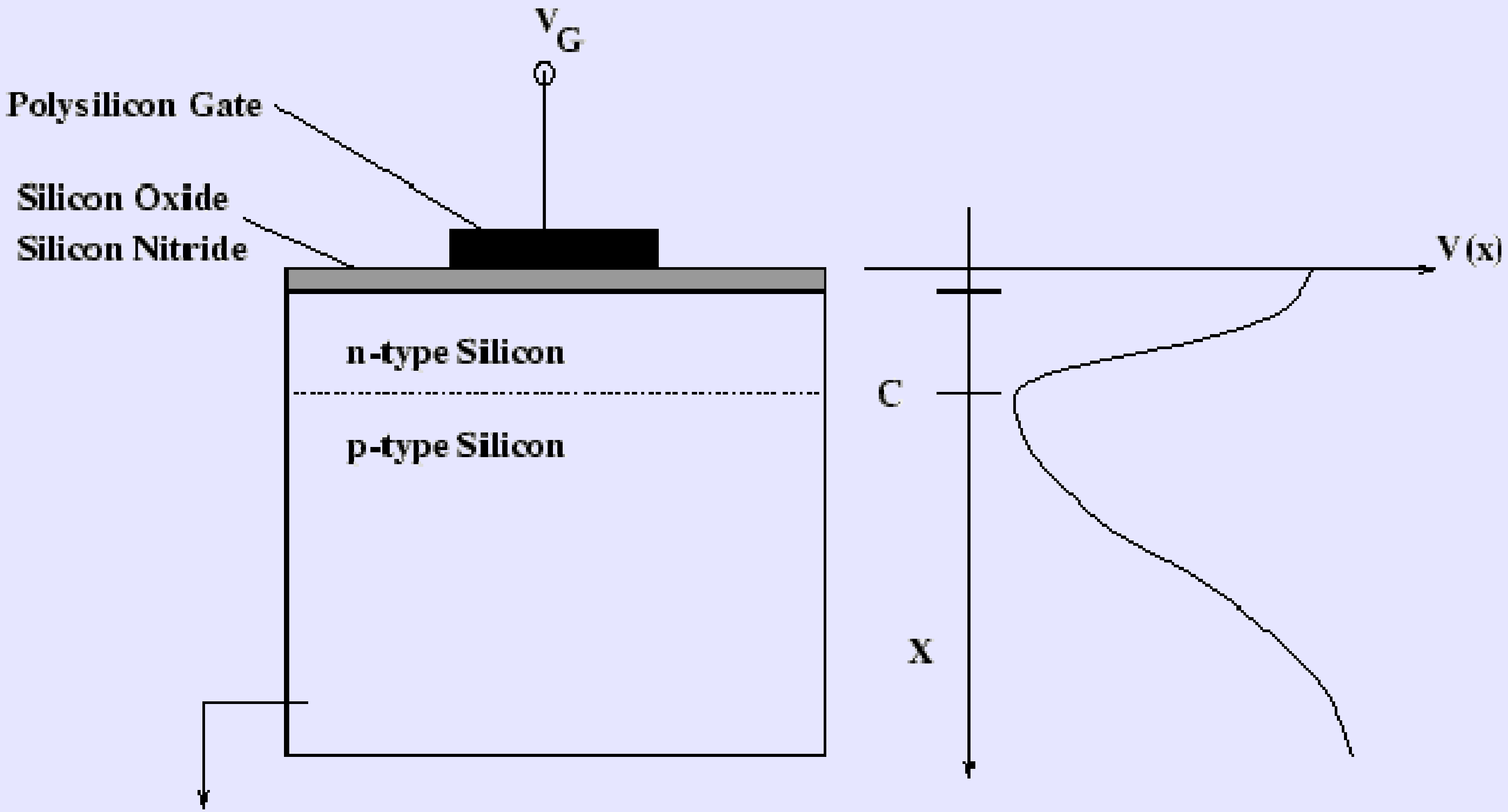
- 8-bit (unsigned) integer bytes
- 16-bit (signed) integers
- 32-bit (signed) integers
- 32-bit single precision floating point real numbers
- 64-bit double precision floating point real numbers

CCDs

Structure

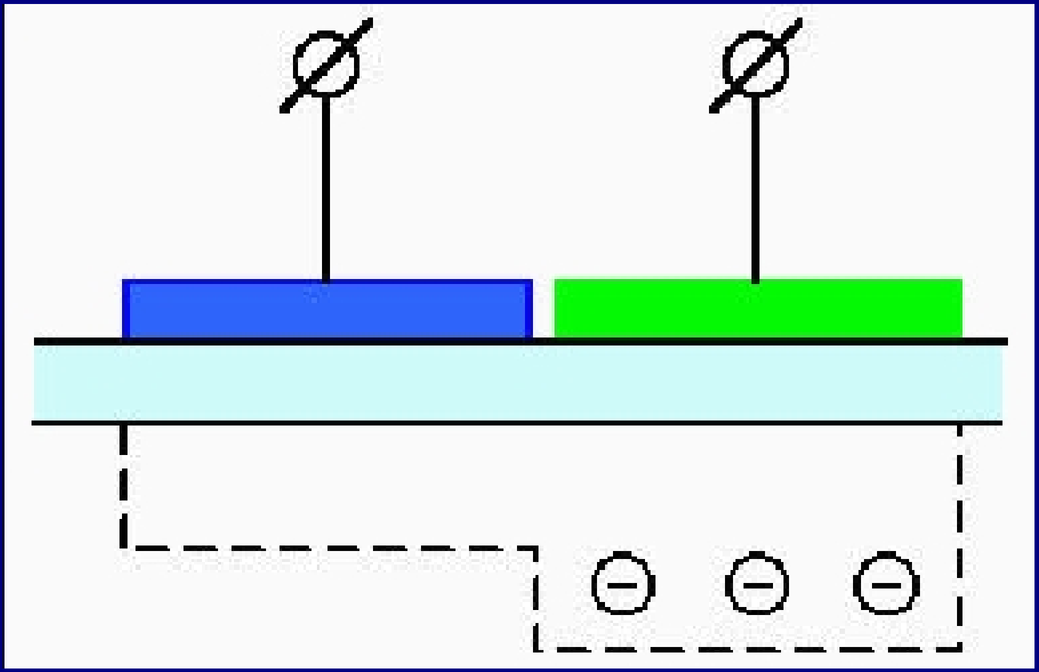
- A charge-coupled device (**CCD**) is an array of metal-oxide-semiconductor (**MOS**) capacitors which can accumulate and store charge due to their capacitance.
- Electrons see a potential well in which they can be trapped. The active region of this device is the depletion region.
- The MOS consists in a electrode (the gate) and a p-type semiconductor, separated by a thin oxide layer (usually SiO_2)
- When a positive voltage is applied to the gate, the induced electric field penetrates through the oxide layer and repels the mobile holes, creating a depletion region.
- Since electrons are attracted to areas of low electrostatic potential, they are attracted to the gate and accumulate in the depletion region near the Si-SiO_2 interface. This region is called a potential well.

MOS capacitors



Adiacent gates

When adjacent gates are close enough their potential wells merge. If the voltage at gate 1 is higher than the voltage at gate 2, the electrons originally in the potential well of gate 2 will travel down into the potential well of gate 1.

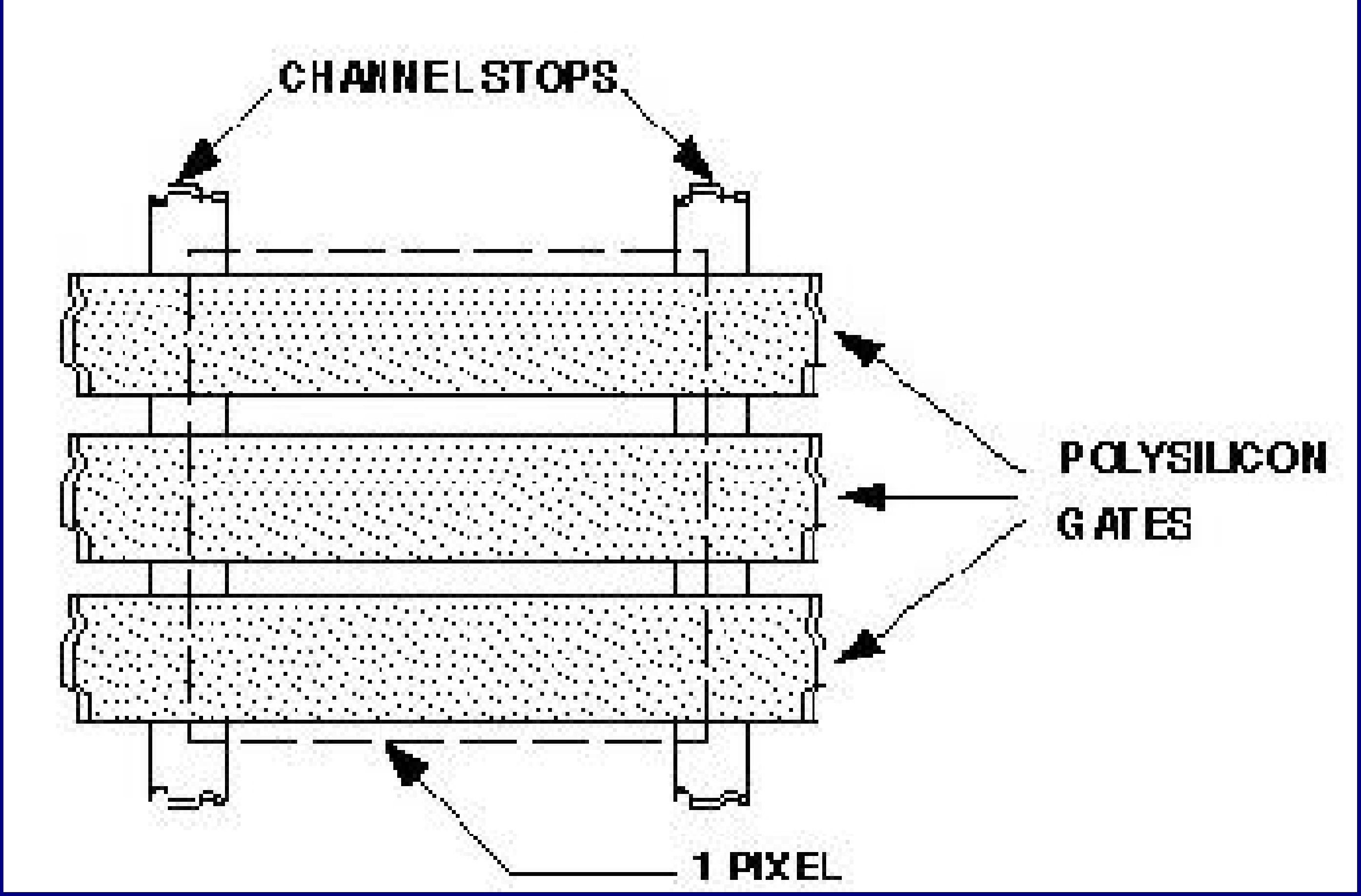


Pixels

A single CCD pixel consists of **three gates** oriented perpendicular to two channel-stop regions.

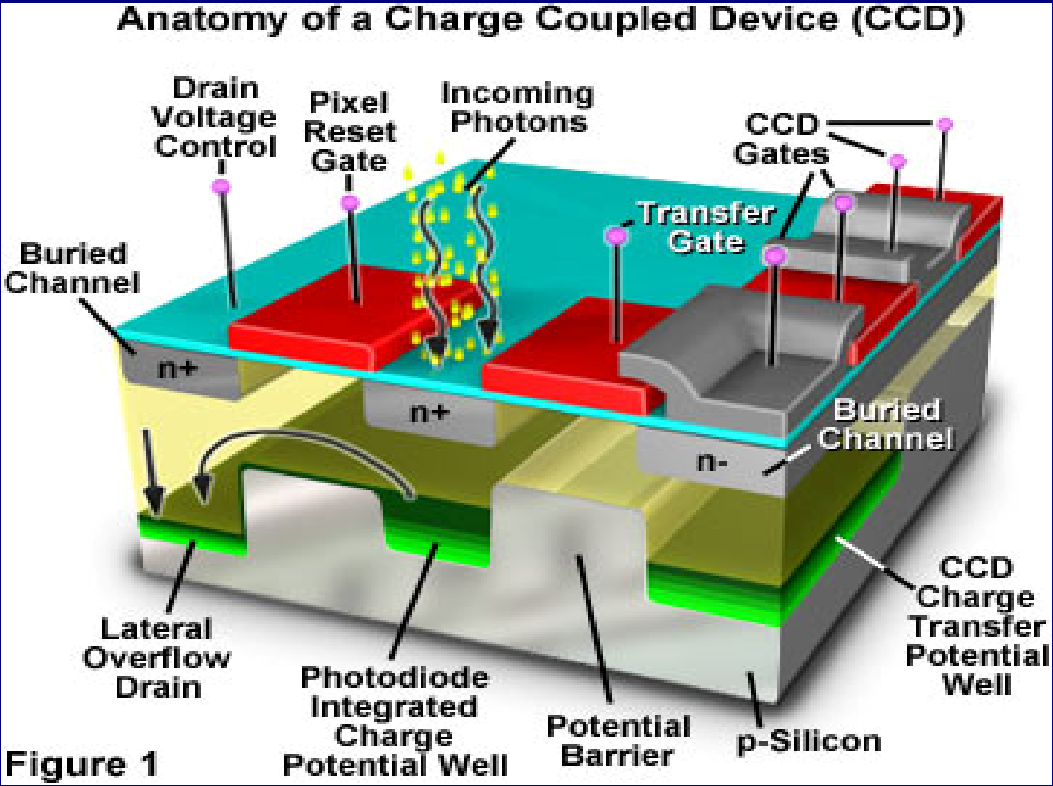
When photons strike a pixel, electron-hole pairs are created via the photoelectric effect.

Top view:



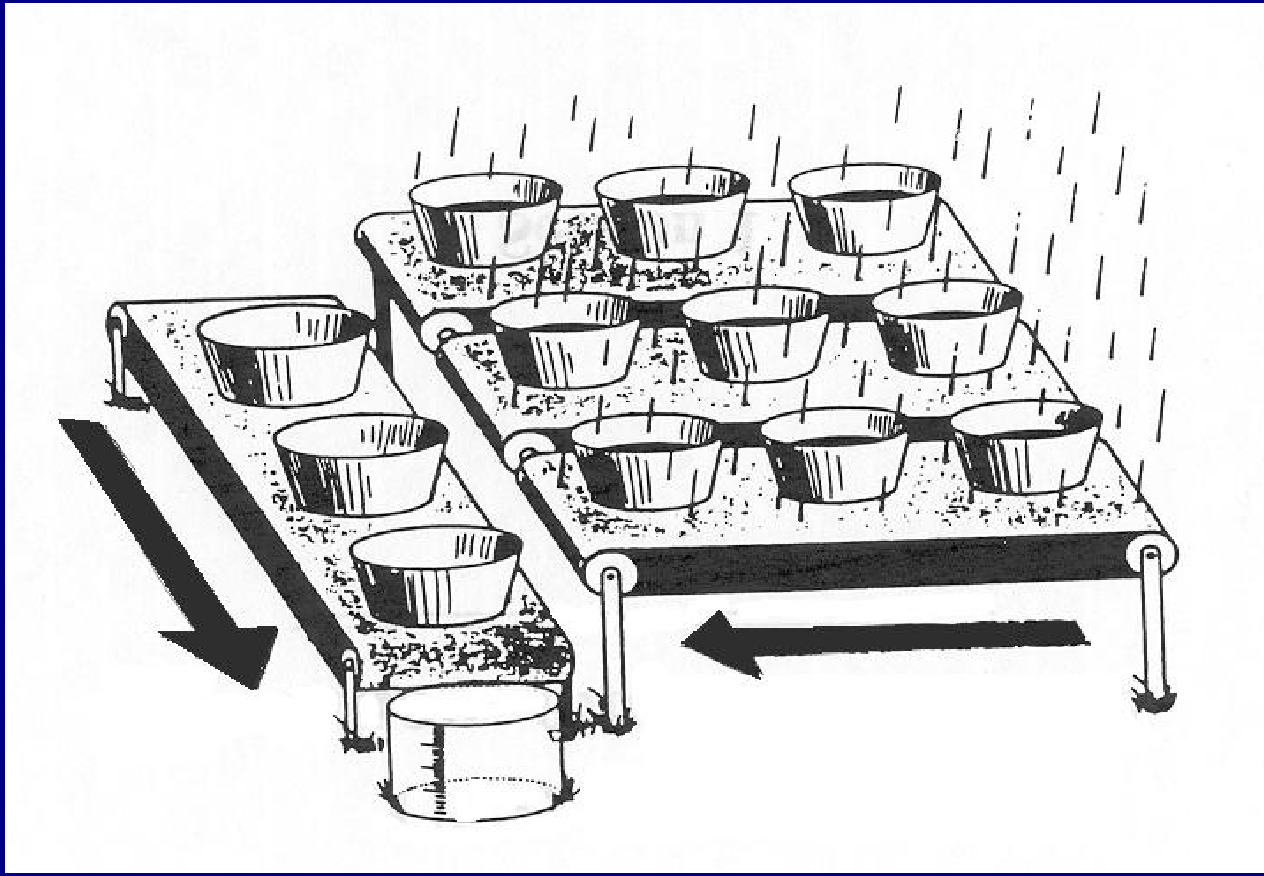
Pixels

- For every electron liberated, a hole is created:
- incident photons create electron-hole pairs.
- The hole, being positive, is repelled by the applied positive potential and escapes into the base of the chip.
- The electron is captured in the nearest potential well



Array Readout

Array readout involves simultaneously clocking each row of pixels, shifting their charge packets one by one towards the serial register. Charge packets are then transferred along the serial register toward the output amplifier where they are detected. The resulting data stream is a pixel-by-pixel representation of the image falling on the CCD array.



Charge transfer

To transfer a charge packet form one pixel to another, three gates are necessary. One to transmit, one to receive, and one to act as a barrier to separate each pair of gates while the charge is being transferred.