Module 3:

Architecture of Special Purpose Computing System

BCSE305L

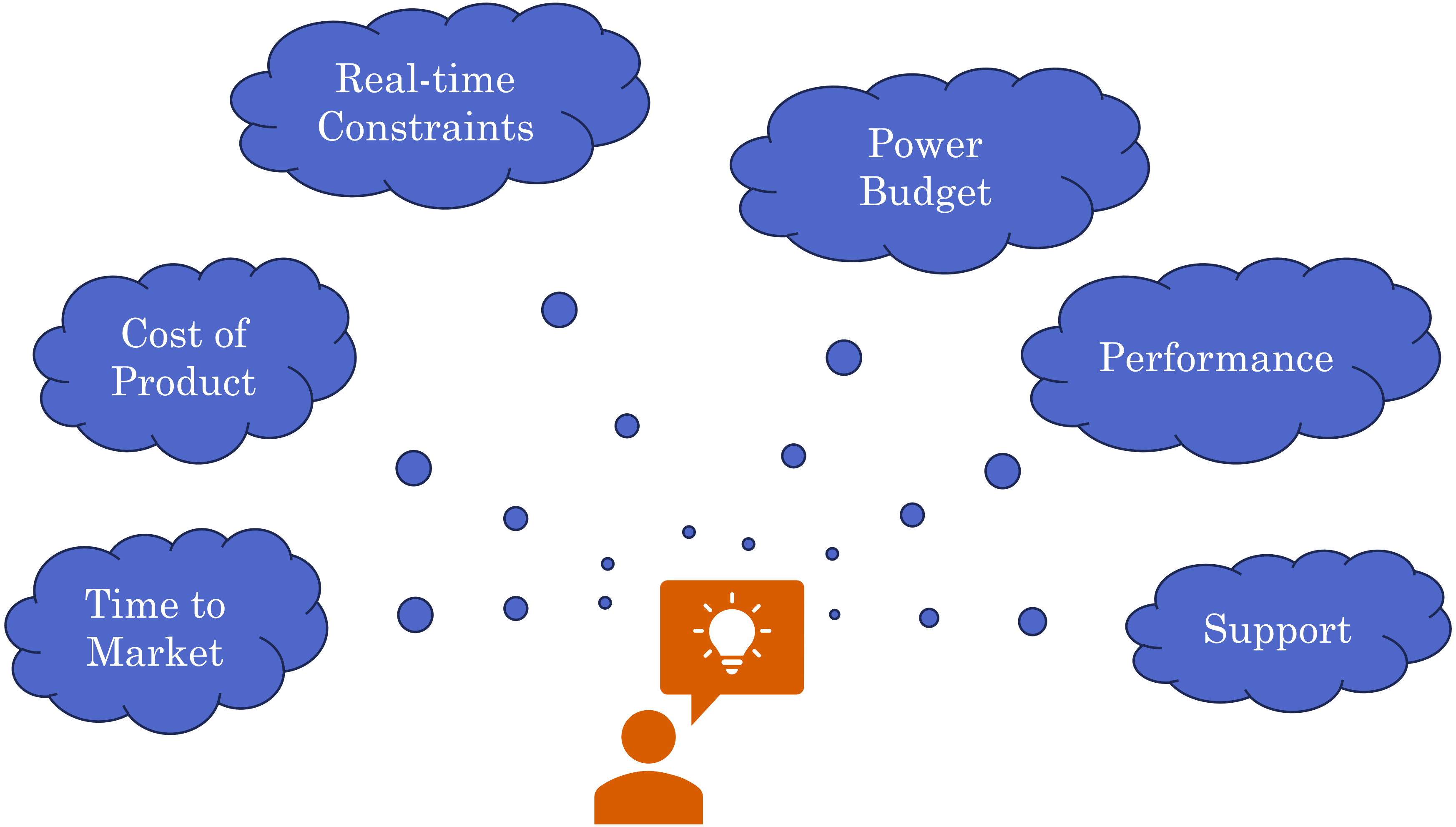
Dr. Vijayakumar P

Outline

- *Handheld devices.*
- *ATM*
- *Data Compressor*
- *Image Capturing Devices*
    - Architecture and Requirements, Challenges.
- *Constraints of special purpose computing system.*

Selecting the **Right Processor**

Selecting the Right Processor



Selecting the Right Processor

- **Is it available in a suitable implementation?**
  - What *good is choosing the highest performing processor* if the *cost of goods* makes your *product noncompetitive in the marketplace*?

- **Is the Processor Capable of Sufficient Performance?**
  - Processor must be *able to do the job on time*.

Selecting the Right Processor

- **Is the Processor Supported by an Appropriate Operating System?**
  - *Porting* the *RTOS kernel* to a *new or different microprocessor architecture* and
  - Having it *specifically optimized* to take advantage of the *low-level performance* features of that microprocessor is not a **EASY** task.

- **Is the Processor Supported by Appropriate and Adequate Tools?**
  - Good tools are critical to project success.
  - At a minimum, you'll need a good cross compiler and good debugging support.

Criteria for **Processor Selection**

Criteria for Processor Selection

- To design an efficient embedded system, ***selection of right processor is very important*** and ***challenging task***

- Types of processors: ***µP***, ***µC***, Digital signal processor (***DSP***)
  - *µP* are offered in **4** to **64-bit size** with distinct features like *cost*, *speed*, *no. of CPU core*, *address* & *data line* are used in *simple toys* to *network router*.
  - *µC* plays an important role in embedded system design and majorly used in *low-end* to *high-end control applications*
  - *DSP* are majorly used for ***high computation intensive applications*** such as *image processing*, *communication devices*, *voice to text converter* etc.,

Criteria for Processor Selection

- Sequence of analysis to be made selecting an appropriate processor for embedded system applications as follows,
  - ***Application requirement analysis***:
    - understand the **purpose** of **application** and arrive specific requirement
  - ***Processor Architecture analysis***:
    - MCS51, ARM, PIC, PowerPC, MIPS etc.,
  - ***Peripheral set analysis***:
    - Includes on-chip (RAM, ROM ,I/O Ports, ADC) and specialized processing units (FPU, MMU, DMA)
  - ***Technical analysis***:
    - Execution *speed*, operating *voltage*, *power* consumption, and *data* & *address bus size* etc.,
  - ***Non-technical analysis***:
    - Cost, software tools, package type, vendor reputation, support etc.,

Criteria for Processor Selection
Case Study 1 : Handheld Device

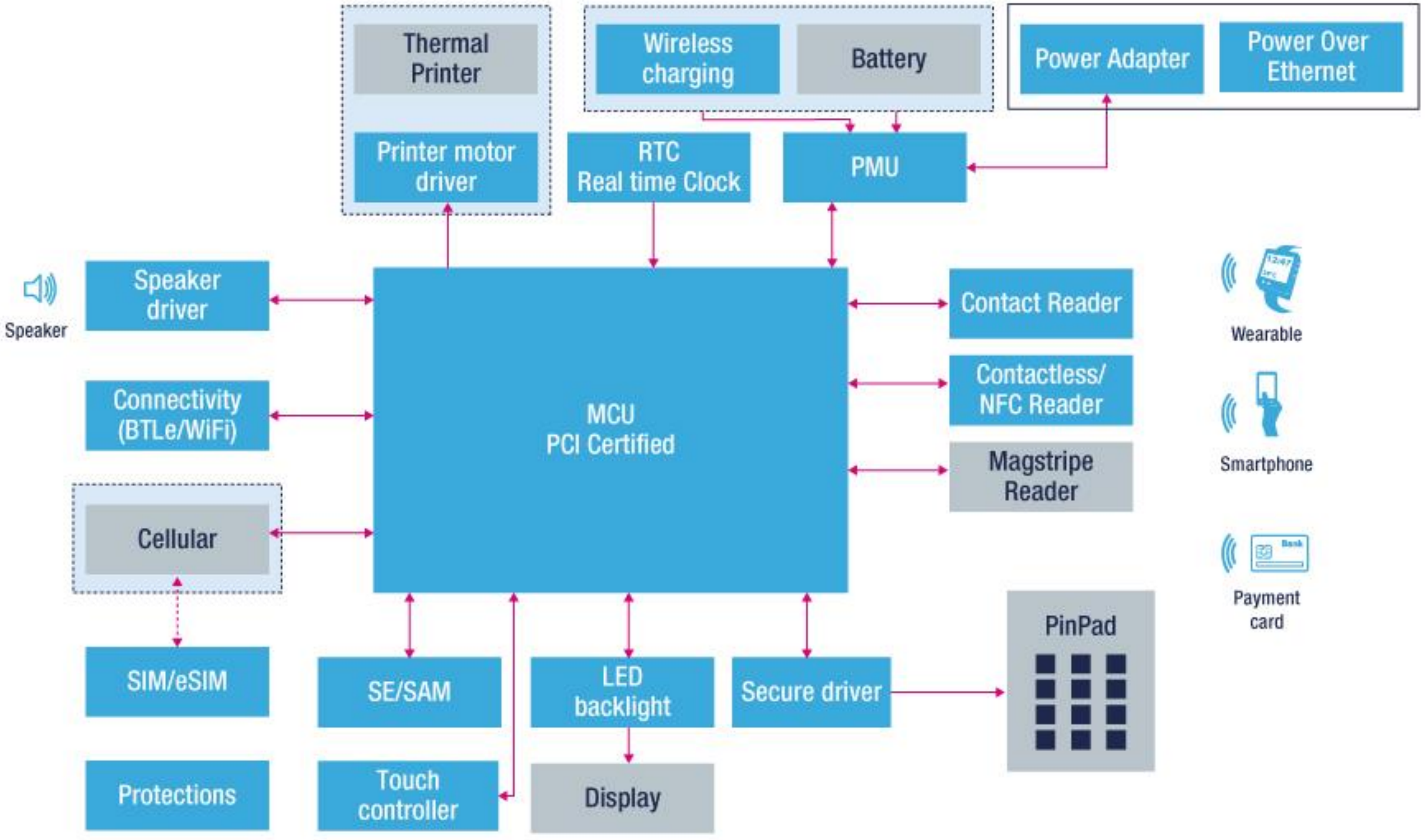
- **Wireless Spot Billing Machine**
  - **POS Hand Held Spot Billing Machine (SBM)** is a **GPRS Enabled** machine.
  - The device is **compact** and **lightweight** (less than 500grms).
  - The machine is <u>*equipped with technology*</u> which serves as a **Hand Held Computer**.
  - The device is *WEB/USB enabled* which helps the *operator* to get *instant bill remotely* and *update transactions back to the server*.

Criteria for Processor Selection
Case Study 1 : Handheld Device

- **Applications of Wireless Spot Billing Machine**
  - Electricity and Water Billing
  - Cable and Internet Billing
  - Canteen and Hotel Billing
  - Finance and Daily Bill Collection
  - Bus and Parking Ticketing
  - Distribution and Retail Billing
  - Petrol and Toll Gate Tokens
  - All kinds of General Billing

Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Block Diagram***



Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Features:***
  - Billing Machine is easy to operate and user-friendly.
  - It is highly secured with Password protection.
  - It is accessible to the remote application via GPRS.
  - It holds the Customer Data in memory.
  - Accurate data Import and Export Settings

Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Specifications:***

Processor	:	**ARM 32-bit Cortex-M3 CPU**(120 Mhz max) with Adaptive real-time accelerator (ART Accelerator), MPU, 150DMIPS/ 1.25DMIPS/MHz (Dhrystone 2.1) capable of **In-System Programming**. **On The Fly Programming** (Can be done in the field using laptop) for Upgrading the units through Serial/USB Ports. **Choice of Program Memory Capacity** of **512K/1024K bytes**.
Memory	:	**8 MB** of **Non-volatile data memory** is provided with the units data retention is minimum 10 years without any power being applied. **Expandable** to **16 MB** **Optional** - It also has **micro SD card** slot for large memory backup.
Real Time Clock	:	In-built **Real time clock** with battery backup.
Key-Board	:	A **35 key multi function key board** is provided on the front panel.
LC Display	:	**132*64 pixels graphical LCD display** with back light provided for user interaction. It has various graphical icons for indication of battery status, signal status etc.

Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Specifications:***

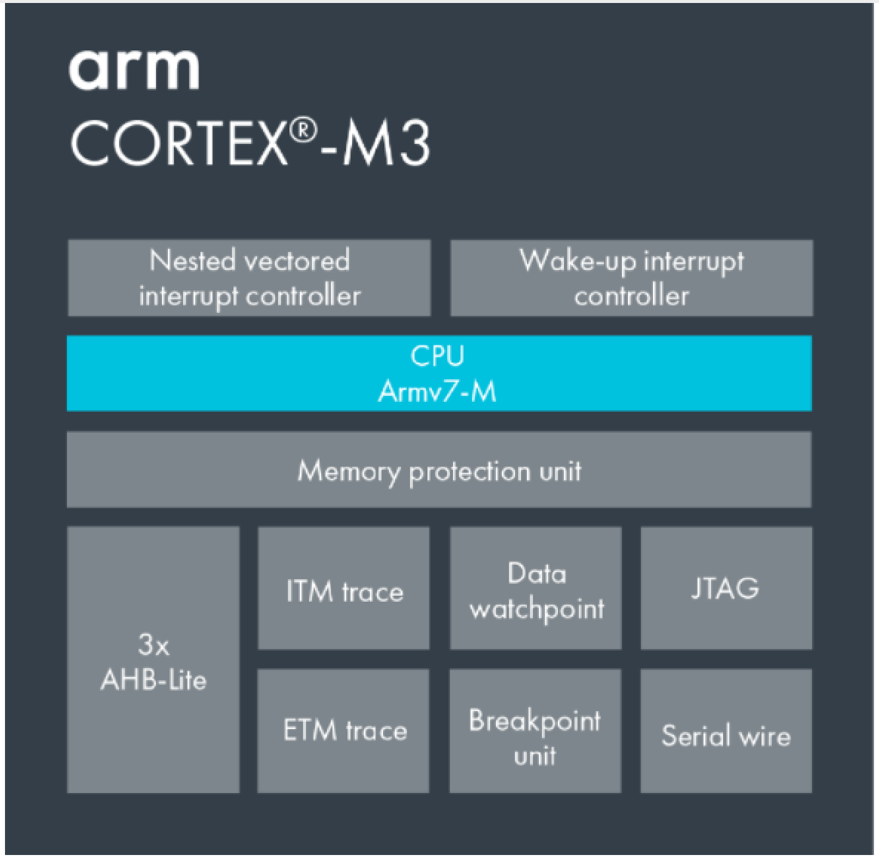
Printer	:	High Speed **24 Column Impact Printer** With **2.7 Lines/Sec**, is Provided as Standard fitment.
Paper Roll	:	57.5mm+/-0.5mm,60mm dia Paper Roll Fitment.
Communication	:	a) **RS232C Ports-1 Nos** (Option of second port) with flexible baud rates. b) Optional IR or IRDA or Both IR or IRDA Port for Communication with electricity meters. c) Built-in **GSM/GPRS** modem optional. d) **Optional GPS**.
Smart Card Interface	:	Optional Contact & Contact less (MIFARE) Smart Card Interface.
Batteries	:	The units are powered by 7.4v 2600 mAH LI-ION Rechargeable Battery pack.
Battery Charger	:	External Universal voltage AC/DC 9.7V-2Amp Adaptor provided.
Mechanical Dimension	:	Width-102.00,Length-276.00,Height-82.00.
Weight	:	Weight-655grams~(without paper roll).

Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Processor***
  - **Cortex-M3** processor is specifically developed for *high-performance*, *low-cost platforms* for a broad range of devices including
    - *Microcontrollers*,
    - *Automotive body systems*,
    - *Industrial control systems* and
    - *Wireless networking and sensors*

Criteria for Processor Selection
Case Study 1 : Handheld Device

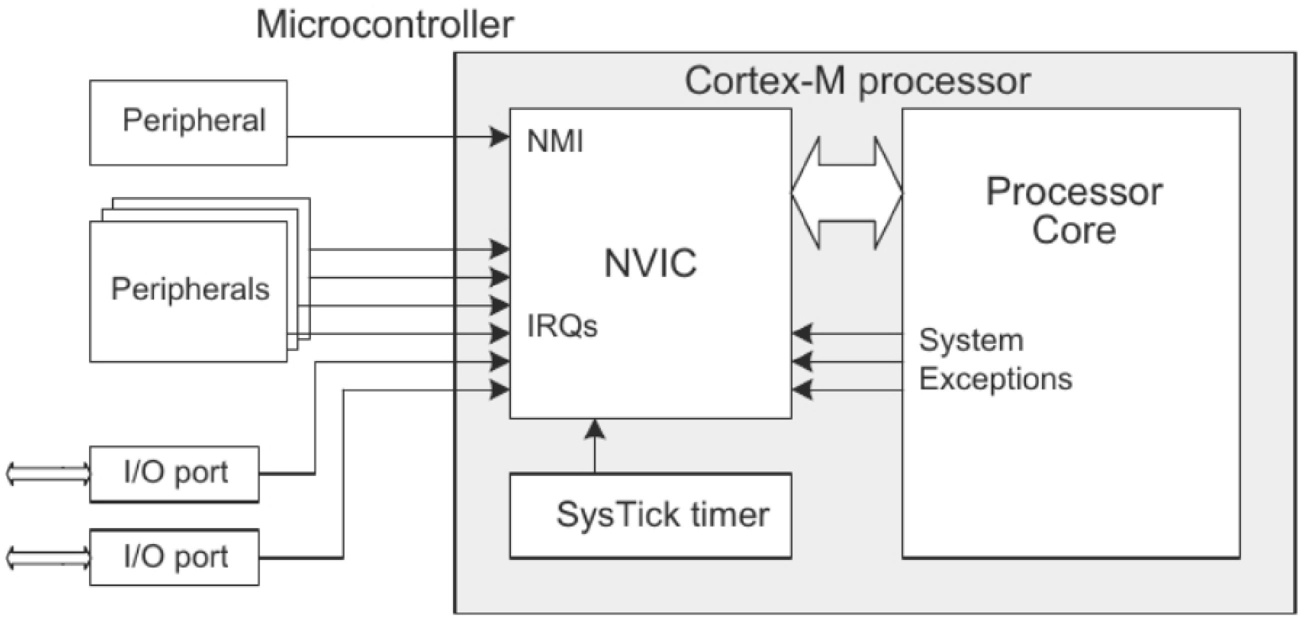
- ***Processor***



Architecture	Armv7-M
Bus Interface	3x AMBA AHB-Lite interface (Harvard bus architecture) AMBA ATB interface for CoreSight debug components
ISA Support	Thumb/Thumb-2 subset
Pipeline	Three-stage
Memory Protection	Optional 8 region MPU with sub regions and background region
Bit Manipulation	Integrated Bit-field Processing Instructions and Bus Level Bit Banding
Interrupts	Non-maskable Interrupt (NMI) + 1 to 240 physical interrupts
Interrupt Priority Levels	8 to 256 priority levels
Wake-up Interrupt Controller	Optional
Enhanced Instructions	Hardware Divide (2-12 Cycles), Single-Cycle (32x32) Multiply, Saturated Adjustment Support
Sleep Modes	Integrated WFI and WFE Instructions and Sleep On Exit capability. Sleep and Deep Sleep Signals Optional Retention Mode with Arm Power Management Kit

Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Nested Vector Interrupt***
  - **Nested vector interrupt control** (NVIC) is a method of ***prioritizing interrupts***, improving the ***MCU's performance*** and ***reducing interrupt latency***.
  - NVIC also ***provides implementation schemes*** for ***handling interrupts*** that occur ***when other interrupts are being executed*** or ***when the CPU*** is in the ***process*** of ***restoring*** its ***previous state*** and ***resuming*** its ***suspended process***.



Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Example 2:***
- ***Rugged Handheld Data Terminal***
  - It is built for life ***out in the field***, integrates **Corning Gorilla glass** for display, able to **handle drops, bumps, spills, dust, vibration**.
  - With ***stable wireless connections*** and ***accurate efficient data capture*** options, you can find this easy-to-deploy device a valuable helper to increase productivity in ***logistics***, ***express delivery***, ***warehousing***, ***retail*** and ***industrial manufacture*** etc.

Criteria for Processor Selection
Case Study 1 : Handheld Device

- *Rugged Handheld Data Terminal*
- <u>Specifications</u>

CPU	Qualcomm Quad-core ARM Cortex-A53 1.3 GHz processor
Operating System	Android 10.0 OS
Memory	8GB ROM + 1 GB RAM(16GB+2GB Optional)
SIM Slot	Support Micro SIM card
PSAM Slot	Support PSAM card (SIM 2 card Optional)
Expansion Slot	Micro SD card (Up to 128G)
Display	4.0 inch high resolution (480*800) IPS TFT display
Touch Panel	Industrial-grade capacitive touch panel (supporting operation with gloves)
Camera	8 Mega-Pixel Auto focus with LED flash
Scanner window	Tempered glass
Wireless LAN	Wi-Fi 802.11a/b/g/n/ac ; 2.4G/5G

Criteria for Processor Selection
Case Study 1 : Handheld Device

- *Rugged Handheld Data Terminal*
- **Processor**

- **Cortex-A53 processor is a high efficiency processor that implements the Armv8-A architecture.**

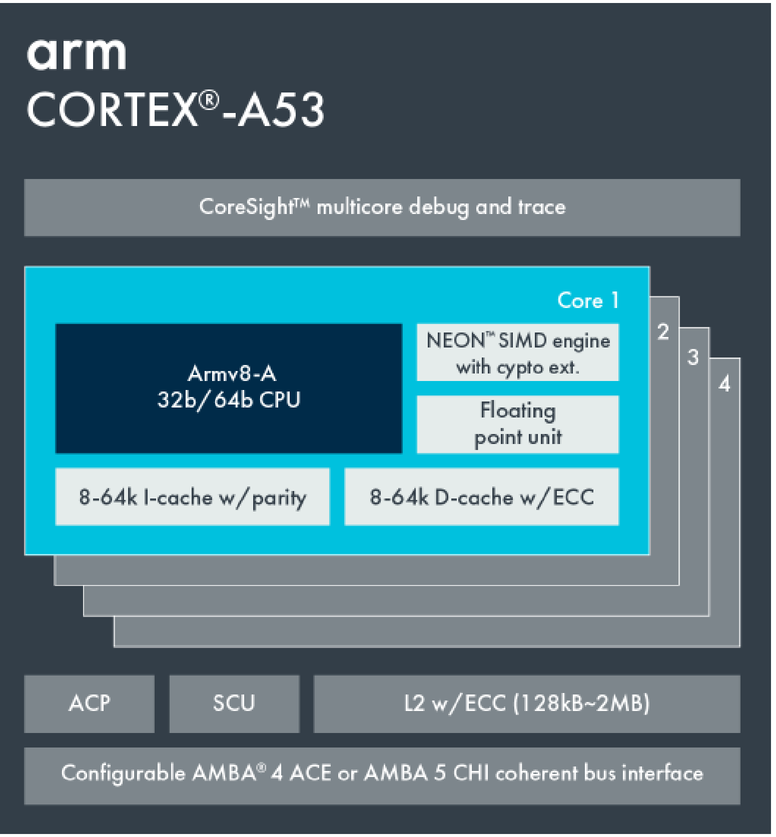
- **The Cortex-A53 processor has one to four cores, each with an L1 memory system and a single shared L2 cache.**

- **It can be combined with other Cortex-A CPUs in a big.LITTLE configuration**

Big.LITTLE configuration refers to a CPU design that features two core clusters with differently-designed cores
High performance cores: These cores are designed for demanding tasks.
Power-efficient cores: These cores handle more conventional tasks.

Criteria for Processor Selection
Case Study 1 : Handheld Device

- ***Rugged Handheld Data Terminal***
- **Processor**



Architecture	Armv8-A
Multicore	1-4x Symmetrical Multiprocessing (SMP) within a single processor cluster, and multiple coherent SMP processor clusters through AMBA 4 technology
ISA Support	• AArch32 for full backward compatibility with Armv7 • AArch64 for 64-bit support and new architectural features • TrustZone security technology • Neon advanced SIMD • DSP & SIMD extensions • VFPv4 floating point • Hardware virtualization support

Criteria for Processor Selection
Case Study 2 :

- **Digital Camera**
  - Digital still cameras require a **significant amount** of **silicon contents**, including the **sensor** (***CCD*** or ***CMOS***), the **analog components** (***ADC***, ***NTSC encoder***, …) and the **engine** (***Digital Signal Processor***).
  - **DSP** - is the brain of the camera and is responsible for ***performing all the computations*** needed to **process** and **compress the image**.

Criteria for Processor Selection
Case Study 2 : Digital Camera

- Digital Camera

Blooming and Streaking Artifacts in CCD Images

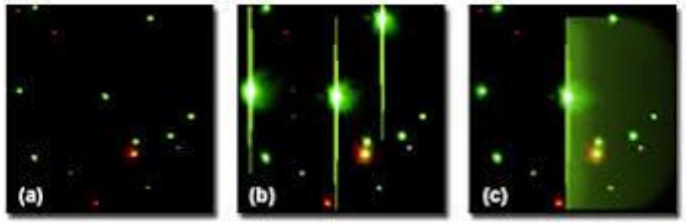
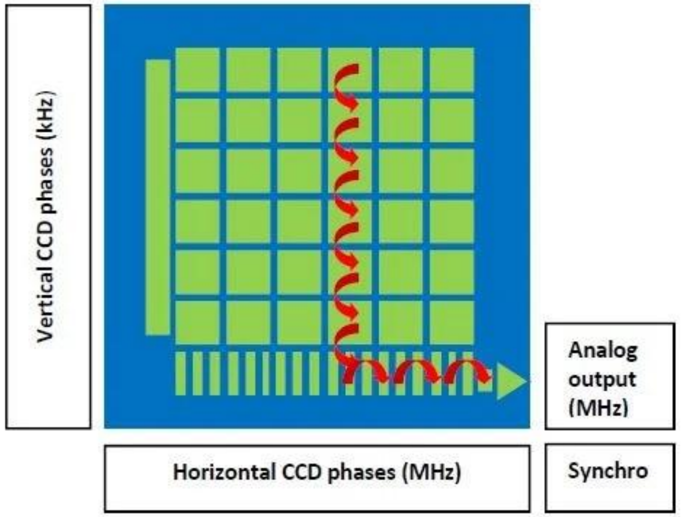
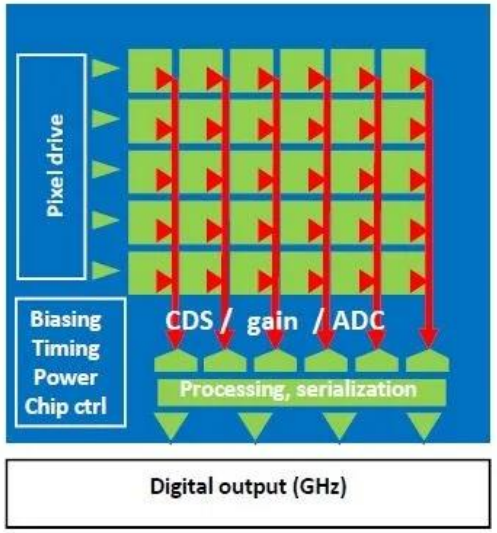


Figure 1



CCD = Photon-to-electron conversion (analog)



CIS = Photon-to-Voltage conversion (digital)

Anatomy of a Charge Coupled Device (CCD)

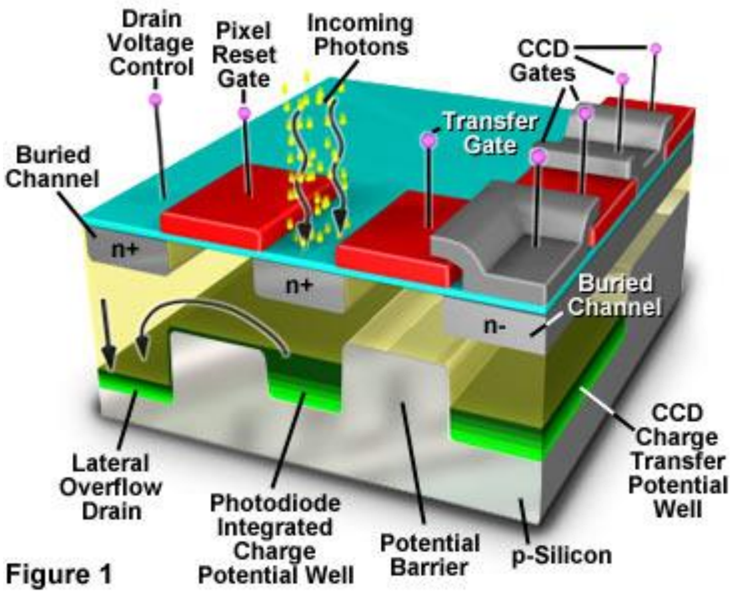
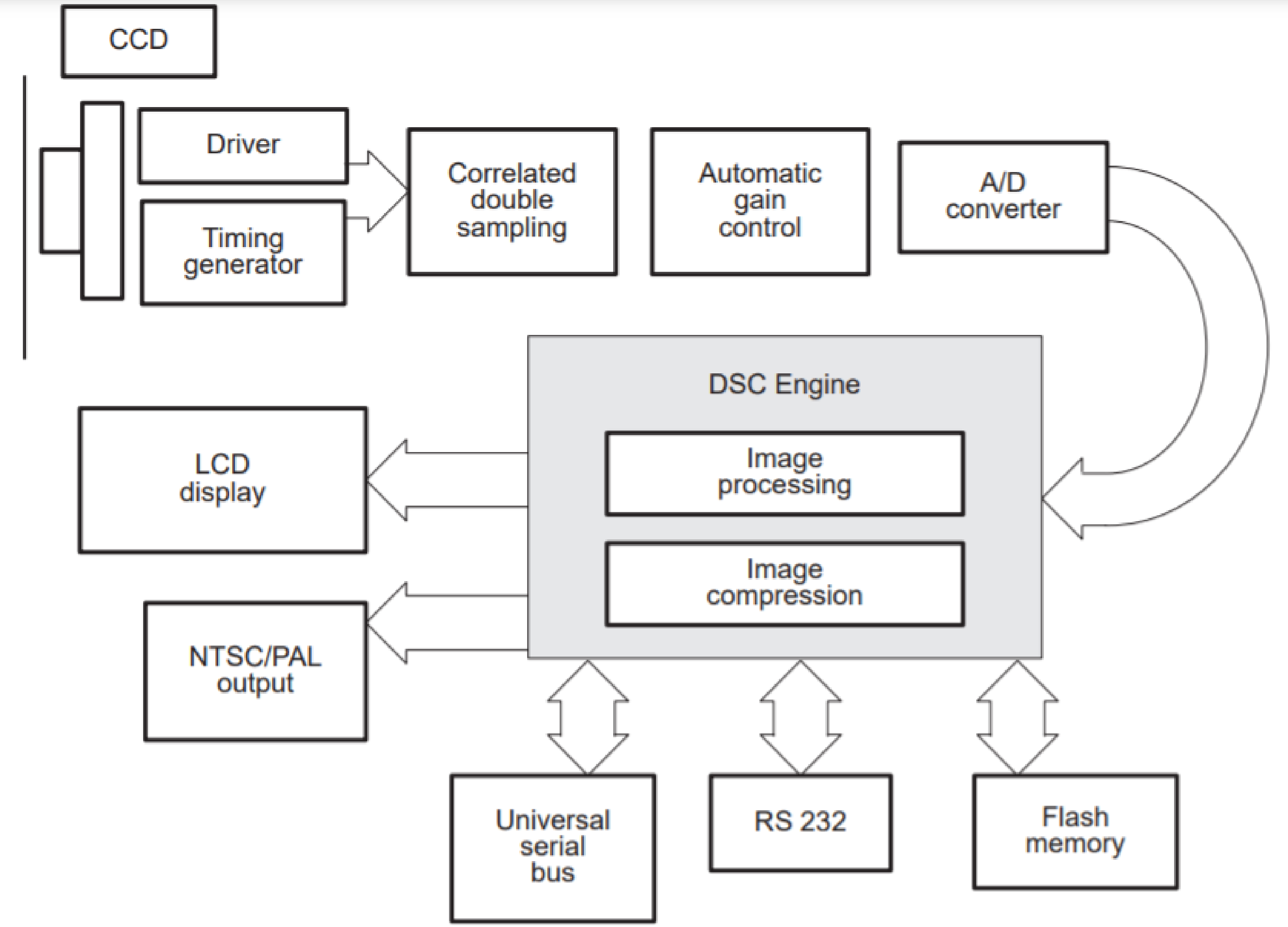


Figure 1

Criteria for Processor Selection
Case Study 2 : Digital Camera

- Digital Camera



Criteria for Processor Selection
Case Study 2 : Digital Camera

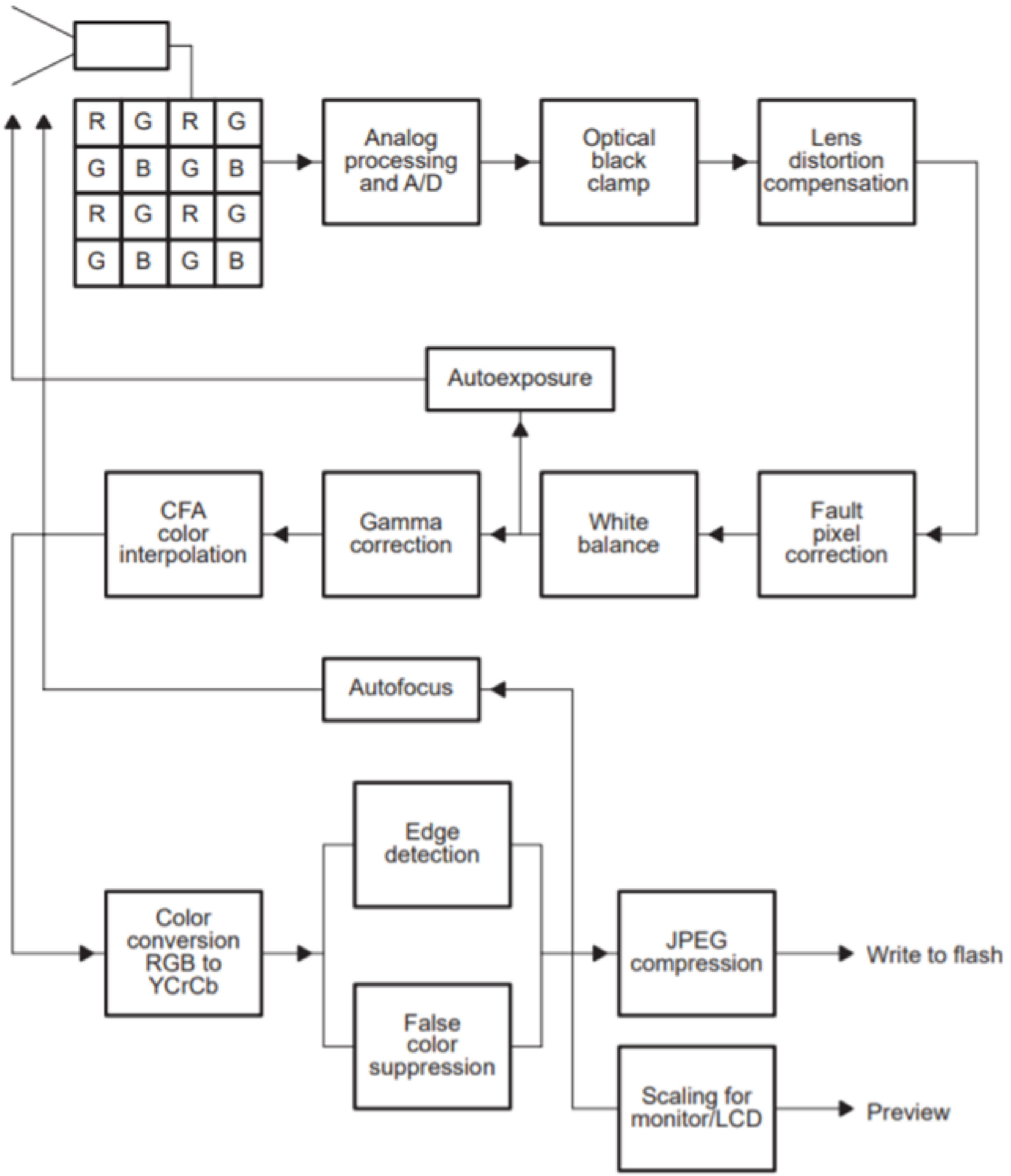
- **Digital Camera**
  - Most DSCs use a **CCD imager** to sense the images.
  - The **driver electronics** and the **Timing Generator** circuitry generate the **necessary signal** to **clock** the **CCD**.
  - **Correlated Double Sampling** and **Automatic Gain Control** electronics are used to get a **good-quality image signal** from the CCD sensor.
  - This **CCD data** is then **digitized** and **fed** into the **DSC engine**.
  - All the image-processing and image-compression operations are performed in the DSC engine.

Criteria for Processor Selection
Case Study 2 : Digital Camera

- **Digital Camera**
  - On most DSCs, the user has the ability to **view the image** to be captured on the **LCD display**.
  - The **compressed images** are stored in **Flash memory** for later use.
  - Most DSC systems also provide an NTSC/PAL video signal to view the captured images (also the preview images) on a TV monitor.
  - The current DSCs also provide ways to connect to the external PC or printer through an RS-232 or a USB port.

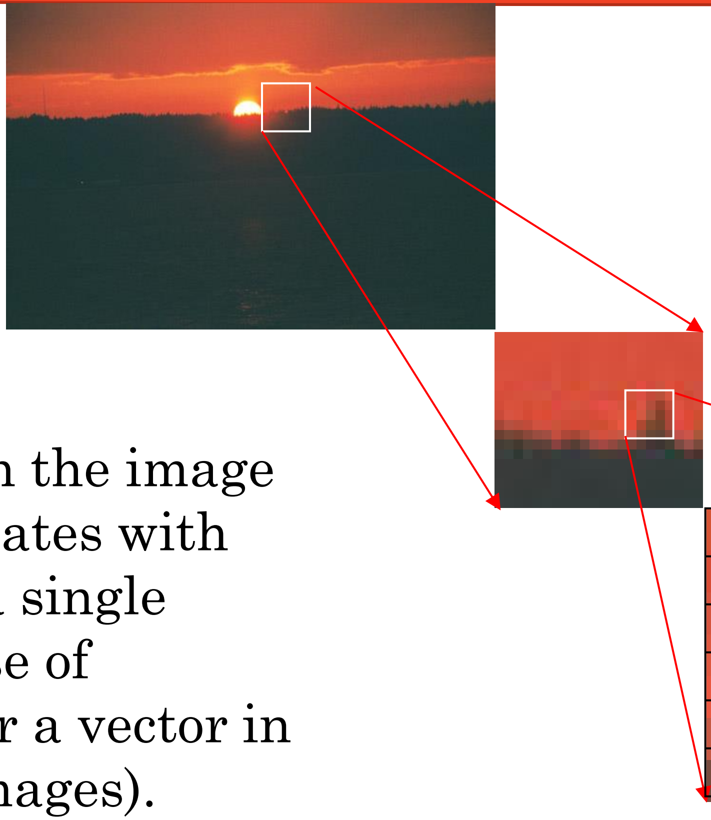
Criteria for Processor Selection
Case Study 2 : Digital Camera

- Digital Camera



Criteria for Processor Selection
Case Study 2 : Digital Camera

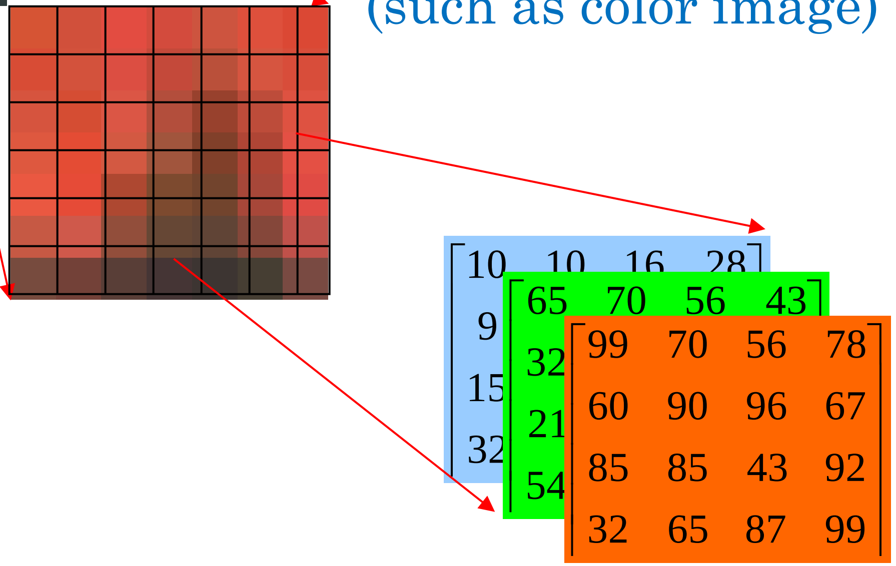
Digital Image



Case study-2: Digital Camera

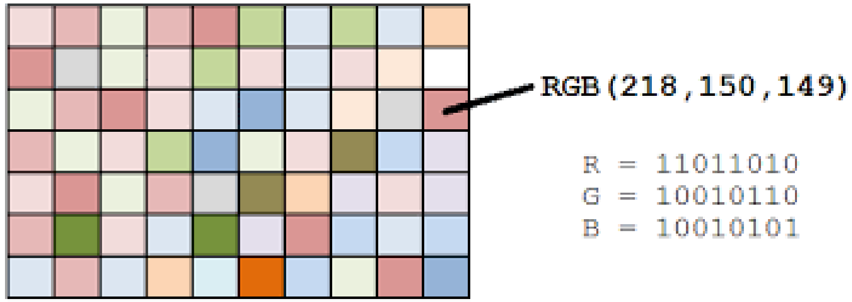
Digital image =
a multidimensional
array of numbers (such as
intensity image) or vectors
(such as color image)

Each component in the image called **pixel** associates with the **pixel value** (a single number in the case of intensity images or a vector in the case of color images).



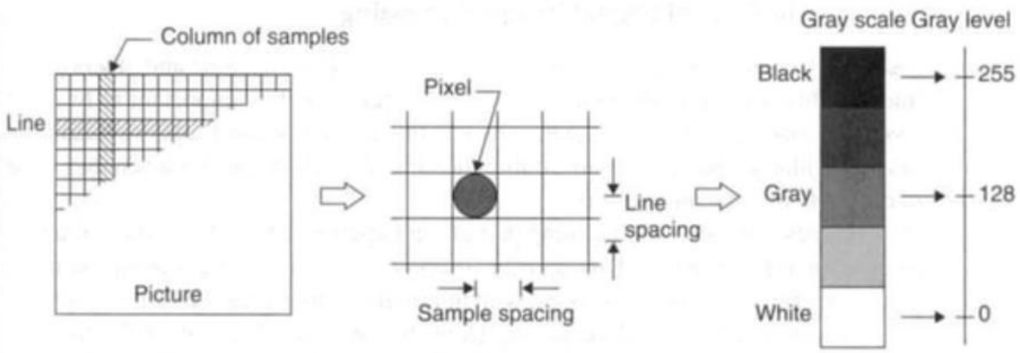
Criteria for Processor Selection
Case Study 2 : Digital Camera

❑ Image Representation



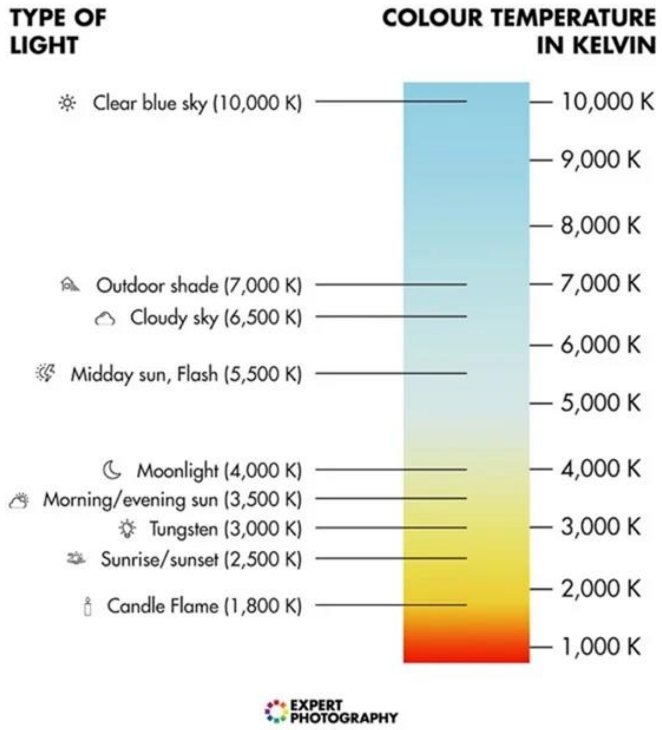
	Red	Green	Blue	Hexadecimal code
	0	0	0	#000000
	255	255	255	#FFFFFF
	255	0	0	#FF0000
	0	255	0	#00FF00
	0	0	255	#0000FF
	255	128	0	#FF8000
	255	255	0	#FFFF00
	128	128	128	#808080

Color	Red	Green	Blue	RGB	Binary
Orange	255	128	0	(255,128,0)	11111111 10000000 00000000
Baby pink	255	210	210	(255,210,210)	11111111 11010010 11010010
Pine Green	0	110	25	(0,110,25)	00000000 01101110 00011001



Criteria for Processor Selection
Case Study 2 : Digital Camera

- Color Balance



Criteria for Processor Selection
Case Study 2 : Digital Camera

- **Image Pipeline**
  - The **Colour Filtered Array (CFA)** data needs to undergo significant amount of ***image processing*** before the image can be ***finally presented*** in a ***usable format*** for ***compression***.
  - All these processing stages are collectively called the "***image pipeline***."
  - Most of these tasks are **multiply-accumulate (MAC) intensive** operations.
  - The TMS320C54x DSP is well suited to perform these tasks efficiently and generate a ***high-quality image*** that is close to the ***image quality*** offered by traditional film from the raw CCD data.

Criteria for Processor Selection
Case Study 2 : Digital Camera

- **Processor - TMS320C54x DSC**
  - On this system, all ***image pipeline operations*** can be ***executed on chip*** since ***only a small 16 x 16 block*** of the image is used.
  - TMS320C549 is well suited due to its **large on-chip memory** (**32K x 16-bit RAM and 16K x 16-bit ROM**).
  - In this way, the processing time is kept short, because there is ***no need for external high-speed memory***.
  - This device offers ***high performance*** (100 MIPS) at ***low power consumption*** (0.45mA/MIPS).

Criteria for Processor Selection
Case Study 2 : Digital Camera

- **Processor - TMS320C54x DSC**
  - Due to the **efficiency** of the *TMS320C54x instruction set* and *architecture*, the *entire image pipeline*, including JPEG, takes about **150 cycles/pixel**.
  - Hence, a **1-Mpixel CCD** image can be processed in **1.5 seconds** on a 100-MHz TMS320C54x.
  - This offers about a **2-second shot-to-shot** delay, including data movement from external memory to on-chip memory.

Criteria for Processor Selection
Case Study 2 : Digital Camera

- ### Processor - TMS320C54x DSC

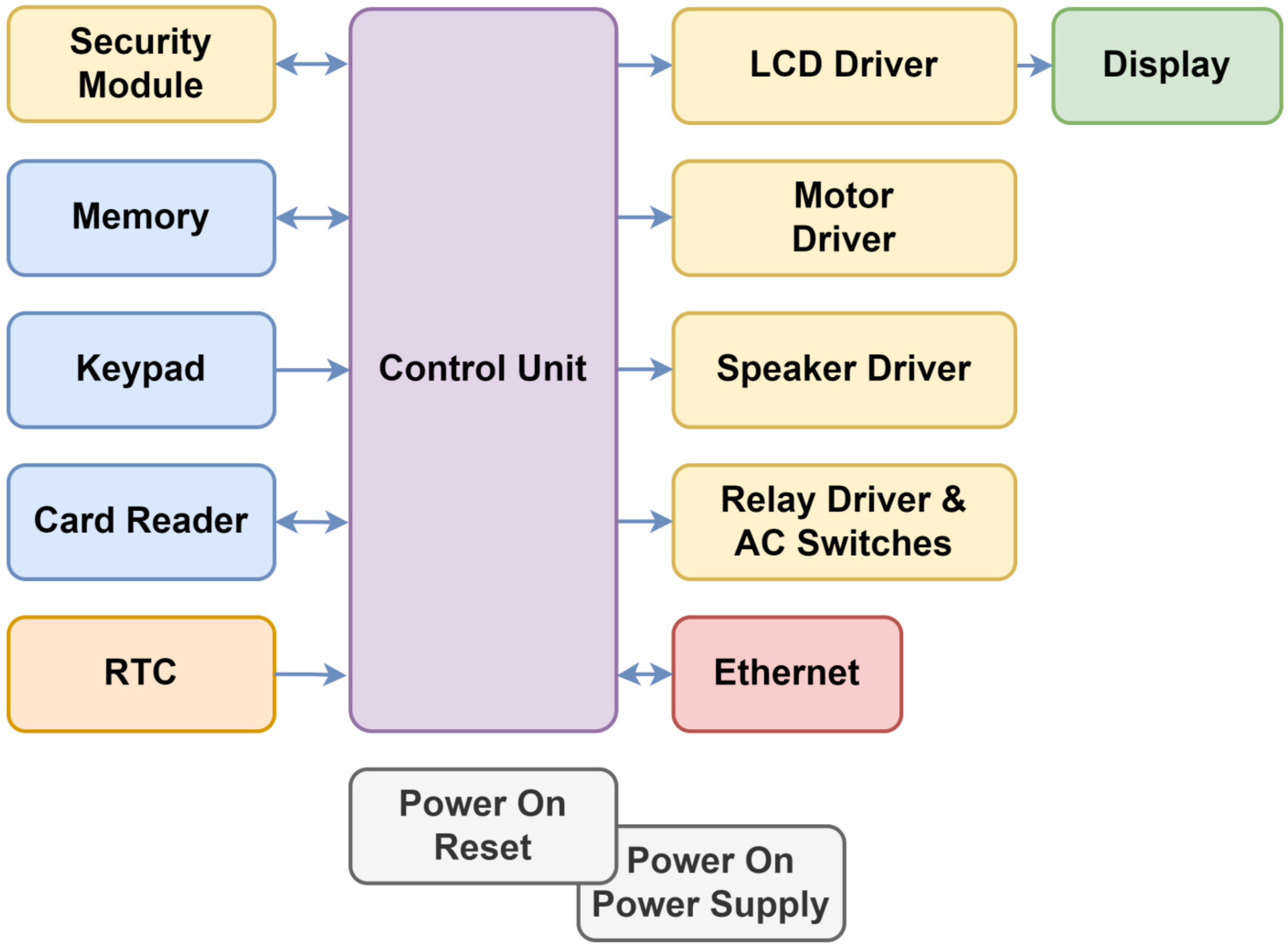
Task	Cycles/Pixel
Preprocessing (Black Clamping, Gain, White Balance, Gamma Correction)	22
Color Space Conversion	10
Interpolation	41
Edge Enhancement & False Color Suppression	27
Total	90
4:1:1 Decimation & JPEG Encoding	62
Total	152

Memory	K Bytes
Program	1.7
Data	4.6

Criteria for Processor Selection
Case Study 3 : ATM

- ### Automatic Teller Machine (ATM)
  - An automated teller machine or automatic teller machine (ATM) is a **computerized telecommunications device** that provides a financial institution's customers a **secure method of performing financial transactions** in a **public space** without the need for **a human clerk or bank teller**.

Criteria for Processor Selection
Case Study 3 : ATM

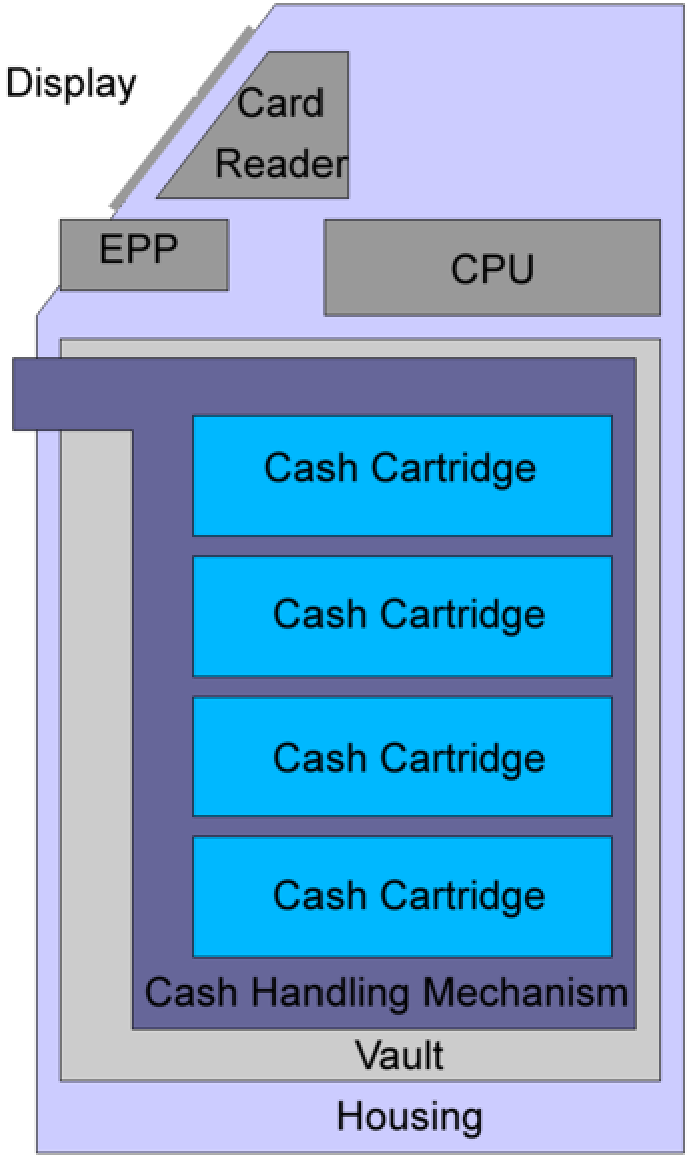


Criteria for Processor Selection
Case Study 3 : ATM

- ## Hardware
  - ***CPU*** (to control the user interface and transaction devices)
  - ***Magnetic*** and/or ***Chip card reader*** (to identify the customer)
  - ***PIN Pad*** (similar in layout to a Touch tone or Calculator keypad), often manufactured as part of a secure enclosure.
  - ***Secure cryptoprocessor***, generally within a secure enclosure.
  - ***Display*** (used by the customer for performing the transaction)
  - ***Function key buttons*** (usually close to the display) or a Touchscreen (used to select the various aspects of the transaction)
  - ***Record Printer*** (to provide the customer with a record of their transaction)
  - ***Vault*** (to store the parts of the machinery requiring restricted access)
  - ***Housing*** (for aesthetics)

Criteria for Processor Selection
Case Study 3 : ATM

- **Vaults**
  - ***Dispensing mechanism*** (to provide cash or other items of value)
  - ***Deposit mechanism*** (to take items of value from the customer)
  - ***Security sensors*** (Magnetic, Thermal, Seismic)
  - ***Locks*** (to ensure controlled access to the contents of the vault)



Criteria for Processor Selection
Case Study 3 : ATM

- **Network**
  - ***Internet service provider*** (ISP) also plays an important role in ATMs.
  - This provides **communication between ATM** and **host processors**.
  - When the ***transaction*** is made, the ***details*** are ***input*** by the ***cardholder***.
  - This ***information*** is passed on to the ***host processor*** by the ***ATM***.
  - The ***host processor*** checks these ***details*** with an ***authorized bank***.
  - If the ***details*** are ***matched***, the ***host processor*** sends the ***approval code*** to the ***machine*** so that the ***cash*** can be ***transferred***.)

Criteria for Processor Selection
Case Study 3 : ATM

- ### Special Purpose Processor
  - Using a ***single-purpose processor*** in an embedded system results in several design metric benefits and drawbacks, which are essentially the inverse of those for general purpose processors.
  - ***Performance*** may be ***fast***, ***size*** and ***power*** may be ***small***, and ***unit-cost*** may be ***low*** for ***large quantities***
  - While ***design time*** and ***NRE costs*** may be ***high***, ***flexibility*** is ***low***, ***unit cost*** may be ***high*** for ***small quantities***, and ***performance may not*** match ***general-purpose processors*** for some applications.

Data **Compression**

Data Compression

- Make ***optimal*** use of ***limited storage space***.

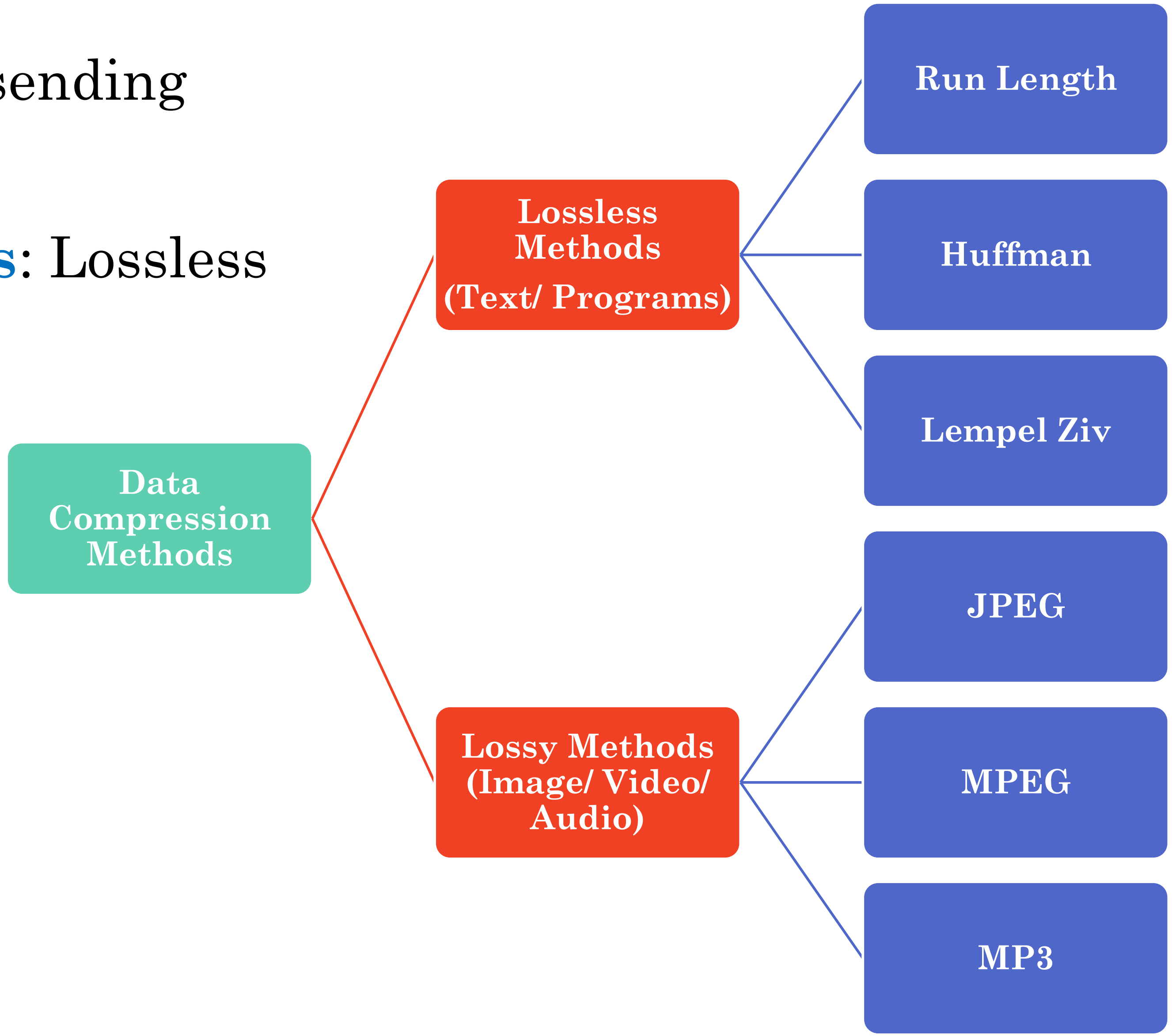
- Save time and help optimize resources.
  - If ***compression*** and ***decompression*** is done in the **I/O processor**
    - *Less time is required to load data to or from the storage system.*
  - In ***sending data*** over ***communication line***
    - *Less time to transmit and less storage to host.*

Data Compression – Entropy

- ***Entropy*** is the *measure* of ***information content*** in a *message*.
  - *A **message** with **higher entropy** carries **more information** than the one with lower entropy*
- ***How to determine the energy of Entropy***
  - Find the probability $p(x)$ of symbol x in the message
  - **Entropy $H(x)$** of the symbol x is:
    - $\boldsymbol{H(x) = -p(x) \cdot \log_i p(x)}$
- *Average Entropy:*
  - Average Entropy over the entire message is the sum of the entropy of all n symbols in a message.

Data Compression – Methods

- It is about storing and sending smaller number of bits.
- **Two major categories**: Lossless and Lossy Methods

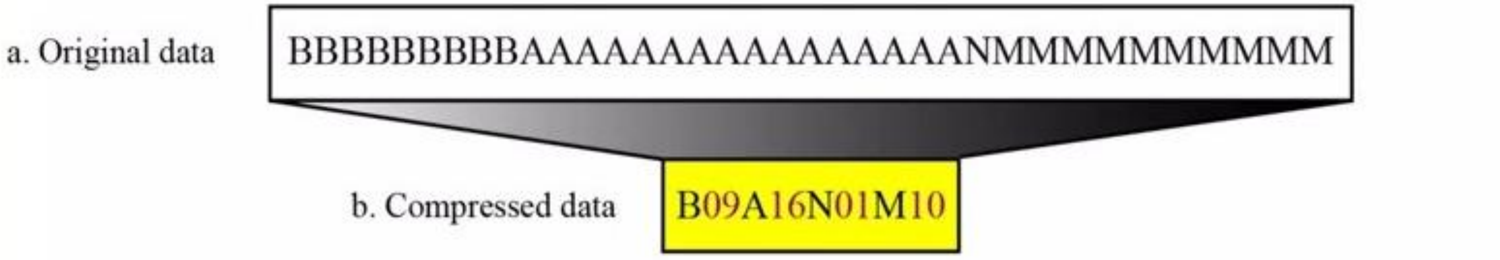


Lossless Compression Methods

- In lossless methods; the **Original Data** and **Data After Compression** & **Decompression** are **exactly the** same.
- **Redundant Data** is **removed** in compression and added during decompression.
- Lossless methods are used in applications, where we can't afford to lose any data:
  - *Legal*
  - *Medical Documents*
  - *Computer Programs*

Run Length Encoding

- *Simplest Form of Compression*
- ***How***: Repeat Consecutive Repeating Sequence by a symbol
  - *Occurrence of the symbols itself*
  - *Followed by the no. of occurrences.*



- The process could be made efficient if the data uses only 2 symbols (0's and 1's) in the bit patterns and 1 symbol is more frequent than the other.
  - 000001111111111000000000000001111110011111 [42 Bits]
    - 0 : 5; 1 : 10 ; 0 : 14 ; 1 : 6 ; 0 : 2 ; 1 : 5
  - 0101110101110011100101101 [25 Bits]

Run Length Encoding

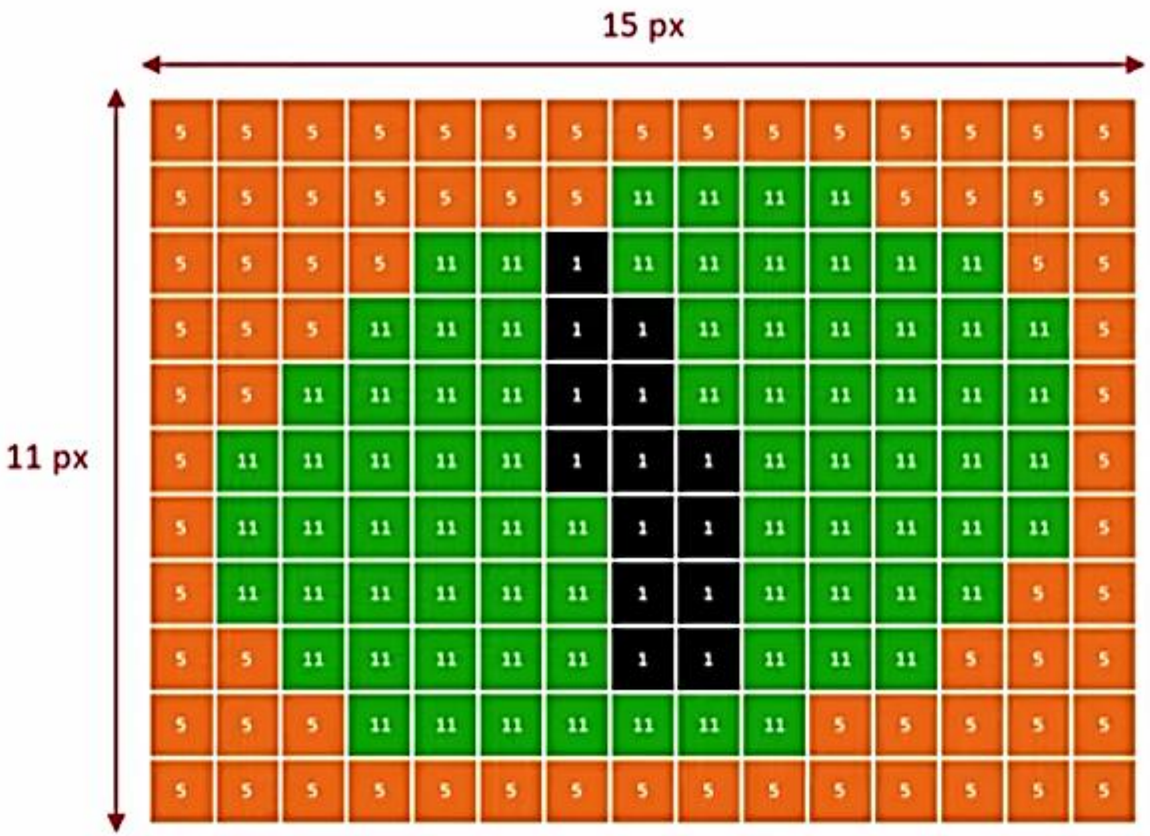
Uncompressed

aaaaabbbbbbbbbbbbccccdddddddddeeeeeeeeee

Uncompressed

aabccdeefghijjjklmnopqrrstttuvvwwxyyyz

Run Length Encoding

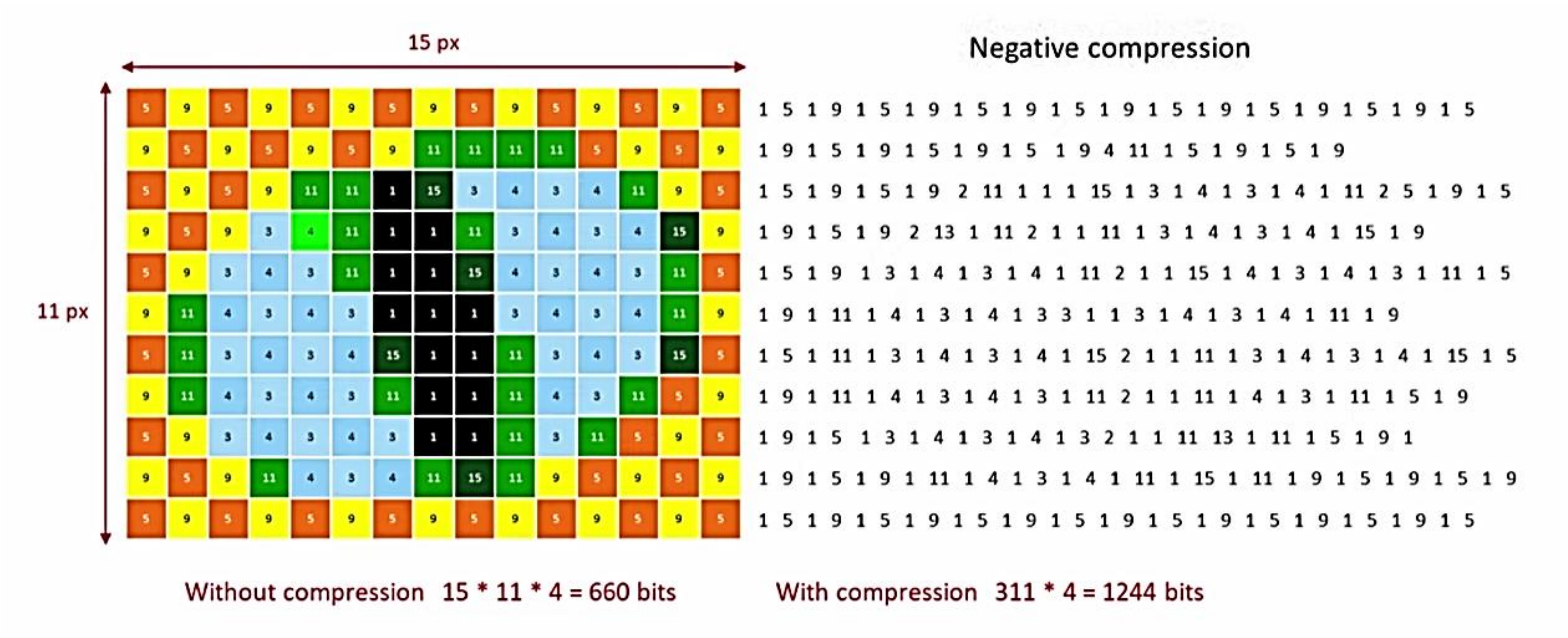


15 5
7 5 4 11 4 5
4 5 2 11 1 1 6 11 2 5
3 5 3 11 2 1 6 11 1 5
2 5 4 11 2 1 6 11 1 5
1 5 5 11 3 1 5 11 1 5
1 5 6 11 2 1 5 11 1 5
1 5 6 11 2 1 4 11 2 5
2 5 5 11 2 1 3 11 3 5
3 5 7 11 5 5
15 5

Without compression 15 * 11 * 4 = 660 bits

With compression 86 * 4 = 344 bits

Run Length Encoding



Huffman Coding

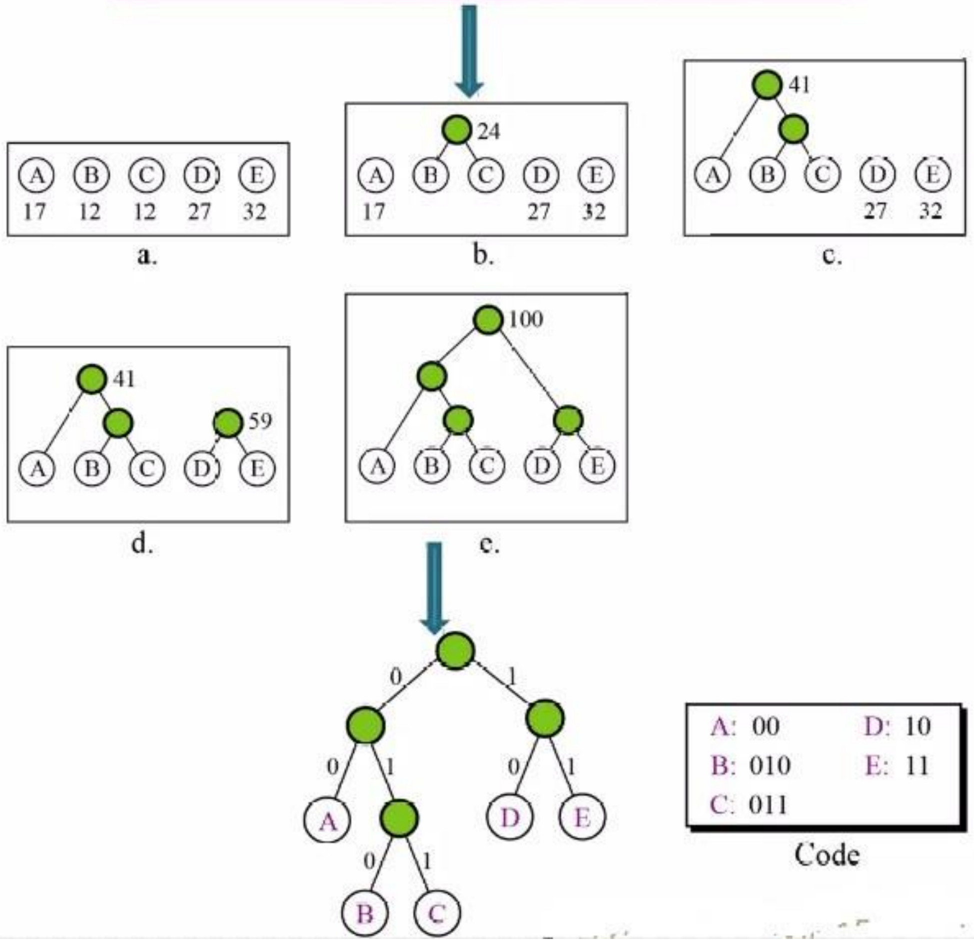
- ***Assign fewer bits*** to ***symbols*** that ***occur more frequently***.
- ***More bits*** to ***symbols*** that ***appear less often***.
- There is **no unique Huffman Code** and every Huffman Code has the same average code length.
- ***Algorithm:***
  - Make a **leaf node** for **each code symbol**
    - Add a ***generation probability*** of each symbol to leaf node
  - Take ***2 leaf nodes*** with ***smallest probability*** and ***connect them*** to a ***new node***
    - Add 1 & 0 to each of the two branches
    - Probability of new node is the sum of the probabilities of two connecting nodes
  - If there is only one node left, the code construction is completed, if NOT, go back to Step 2.

Huffman Coding

- Example

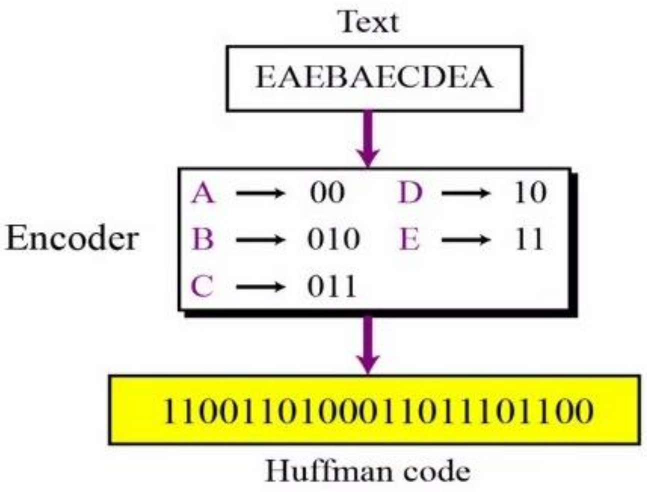
Table 15.1 Frequency of characters

Character	A	B	C	D	E
Frequency	17	12	12	27	32

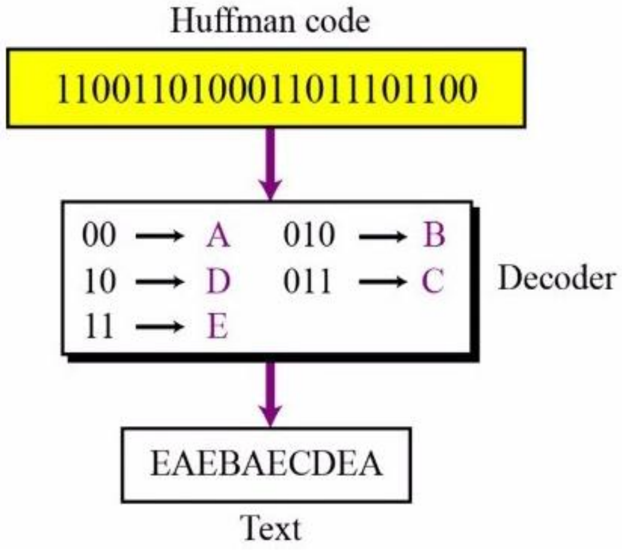


Huffman Coding

- Encoding



- Decoding



Lempel Ziv Coding

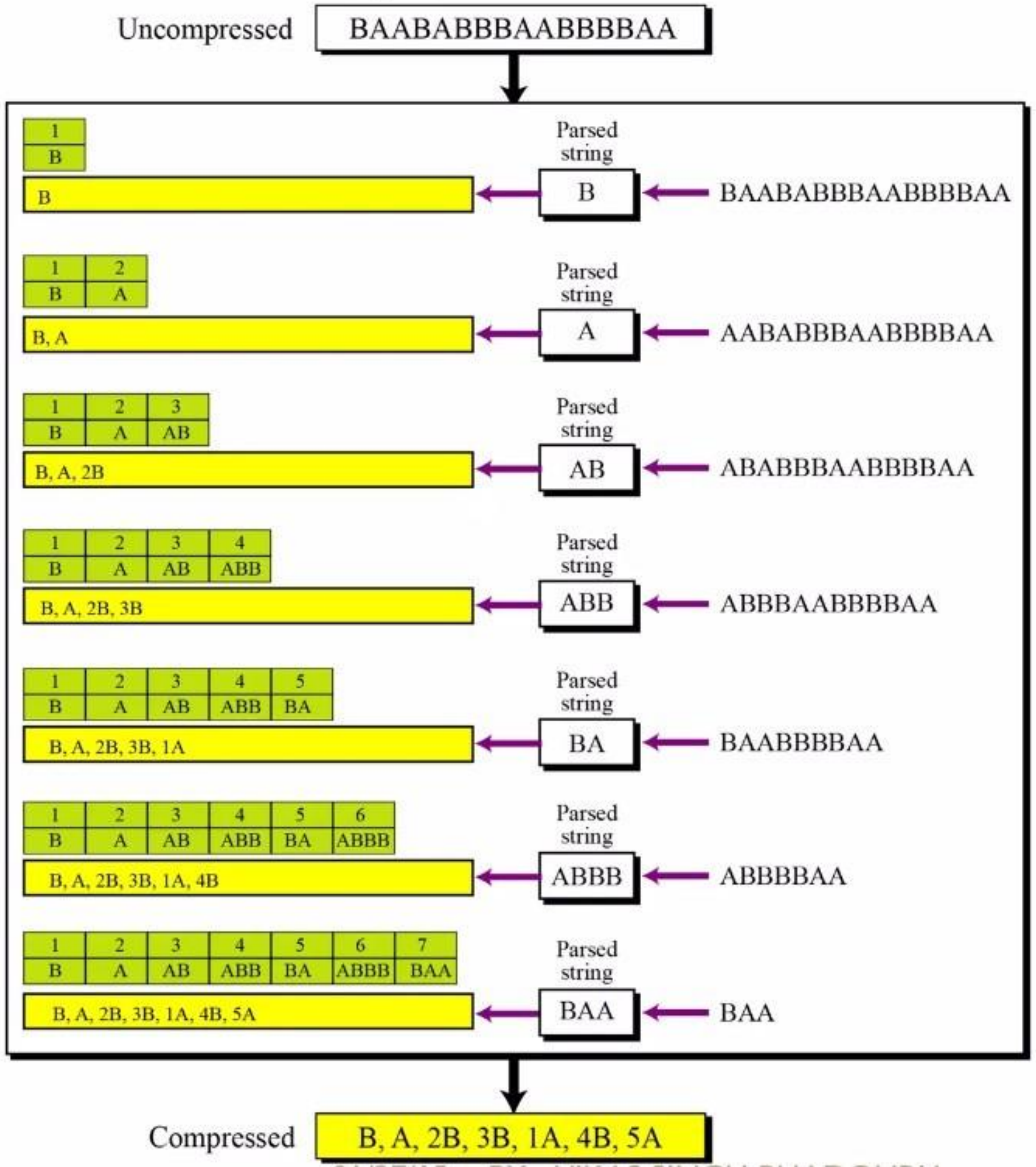
- It is a **dictionary** based **encoding**.
- **Basic idea:**
  - **Create** a **dictionary** (a table) **of strings** used during the communication.
  - If ***both sender*** and ***receiver*** have a ***copy*** of the ***dictionary***, then the **previously-encountered strings** can be **substituted** by **their index** in the dictionary.
- **2 Phases of Operation**
  - ***Building*** an indexed dictionary.
  - ***Compressing*** a string of symbols.

Lempel Ziv Coding

- **Algorithm:**
  - Extract the ***smallest substring*** that *cannot be found* in the *remaining uncompressed string*.
  - *Store* that *substring* in the *dictionary* as a *new entry* and *assign* it an *index* *value*.
  - Substring is replaced with the index found in the dictionary.
  - Insert the index and the last character of the substring into the compressed string.

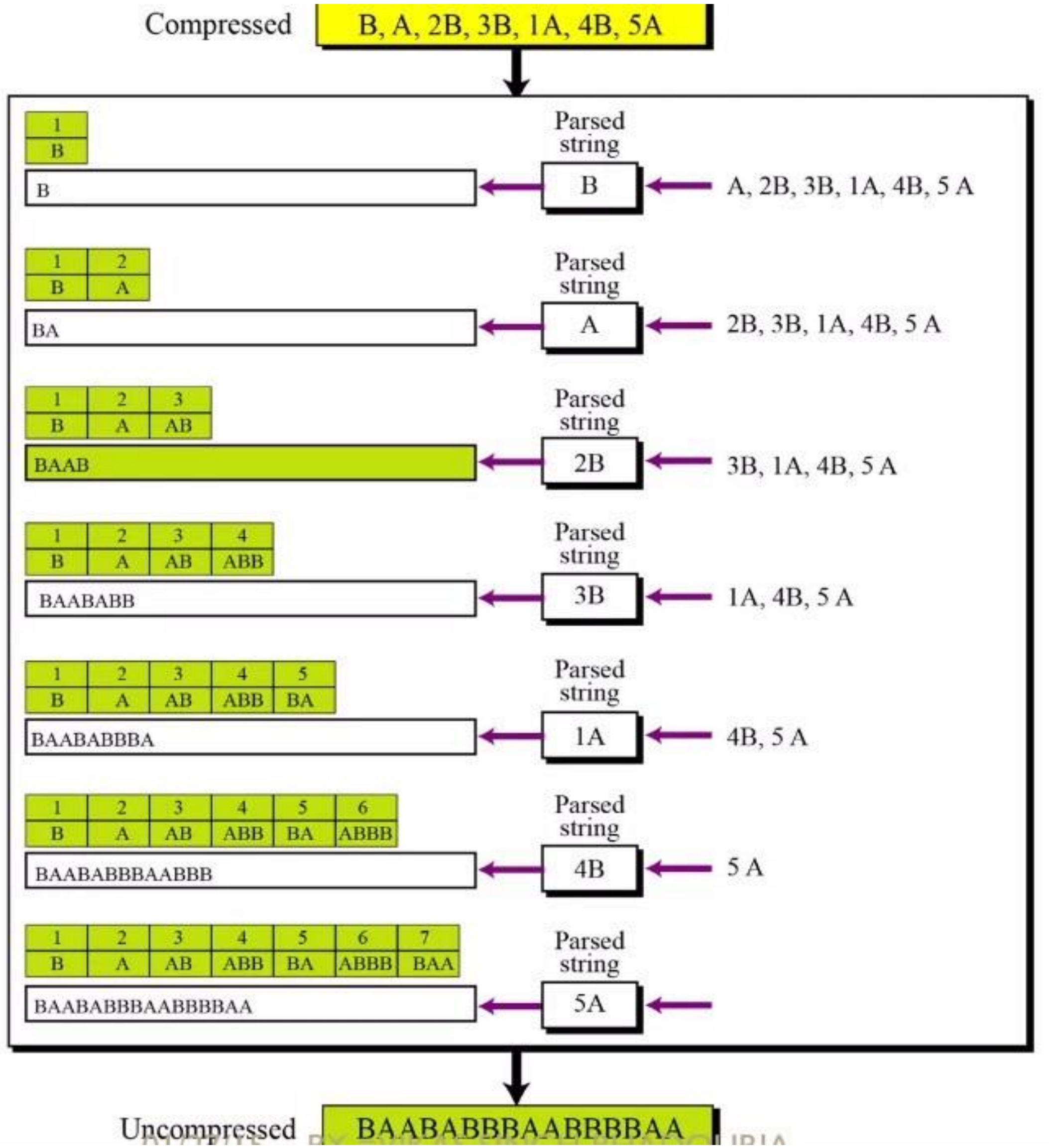
Lempel Ziv Coding

- Compression example:



Lempel Ziv Coding

- It's just the inverse of compression process



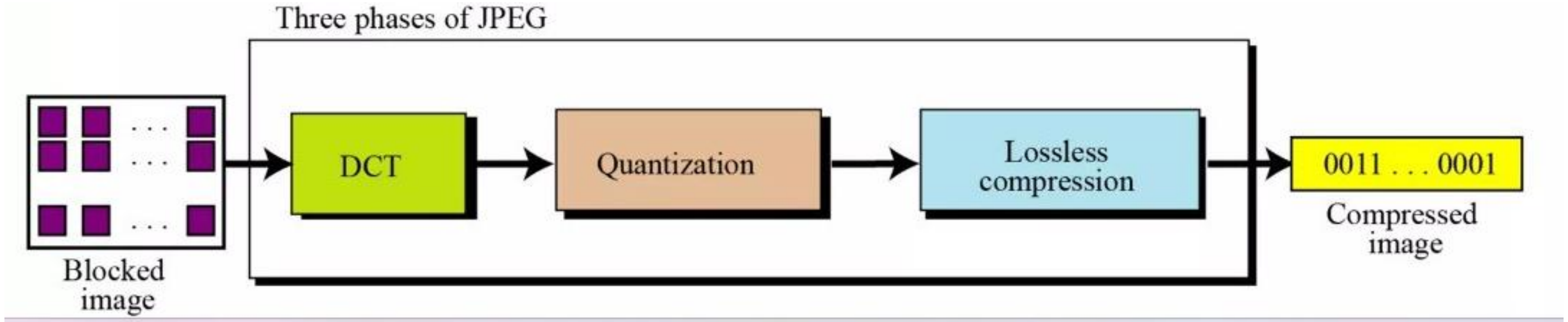
Lossy Compression Methods

- Used for compressing images and video files.
  - *Our eyes cannot distinguish subtle changes.*
  - *So, Lossy Data is Acceptable*

- These methods are cheaper, less time and space dependence.
- Example:
  - JPEG : for compressing pictures / images
  - MPEG : compressing video
  - MP3 : compress audio

JGEG Encoding

- Used for compressing images & graphics
- In JPEG, the grayscale picture is divided into 8 x 8 pixel blocks to decrease the number of calculations:
- Basic Idea:
  - Change the picture into a linear (vector) set of numbers that reveals the redundancies.
  - The redundancies is then removed by one of the lossless methods

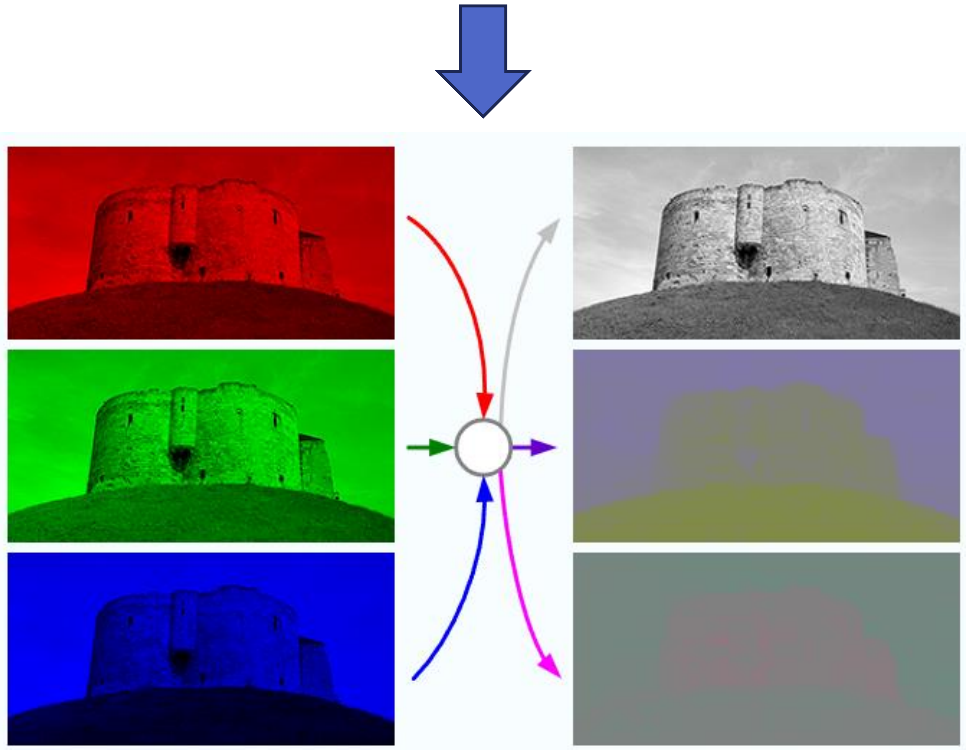


JGEG Encoding

- JPEG, compression based on two psychovisual principles:
  - ***Changes in brightness are more important than changes in colour***:
    - The human retina contains about 120 million brightness-sensitive rod cells, but only about 6 million colour-sensitive cone cells.
  - ***Low-frequency changes are more important than high-frequency changes***.
    - **Human eye** is **good** at **judging low-frequency light changes**, like the edges of objects.
    - **Less accurate** at **judging high-frequency light changes**, like the fine detail in a busy pattern or texture.
      - *Camouflage works in part because higher-frequency patterns disrupt the lower-frequency edges of the thing camouflaged*

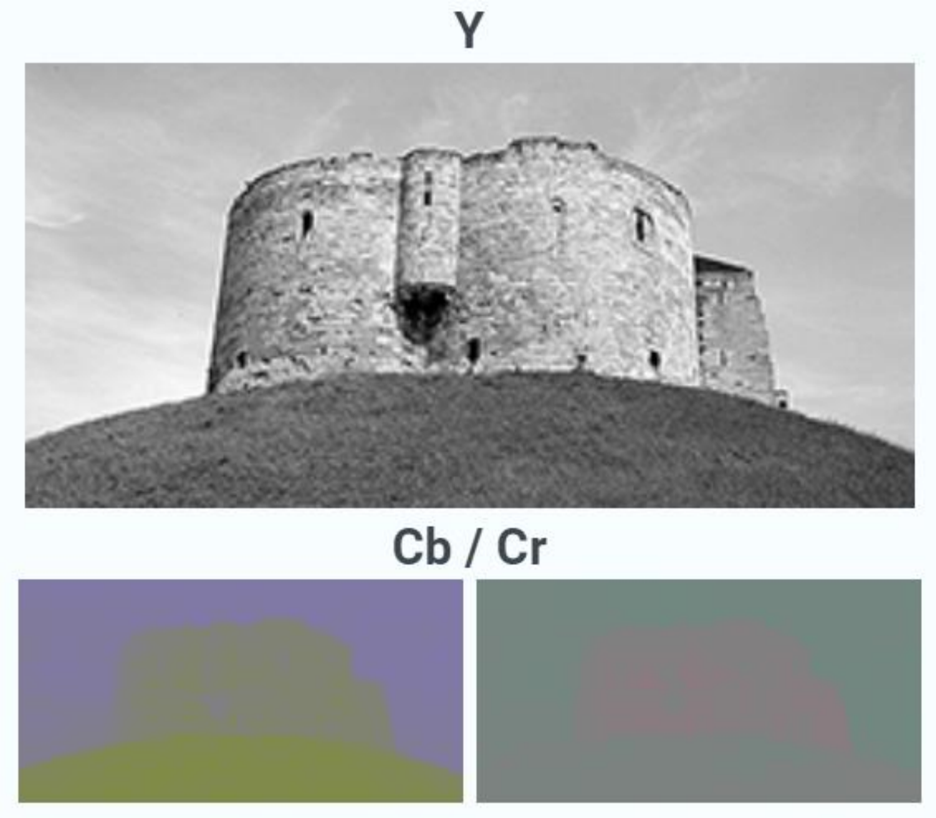
JGEG Encoding

- **Step 1: Isolation of Color Information**
  - Image's **brightness** information is **spread evenly through** the R, G, and B channels.
  - **Isolate brightness** from the **colour information**.
  - To do this, JPEG uses some math to **convert the image's colour space** from **RGB** to **YCbCr**. A YCbCr image also has three channels
    - *But it stores all of the brightness information in one channel (Y) while splitting the colour information between the other two (Cb and Cr).*



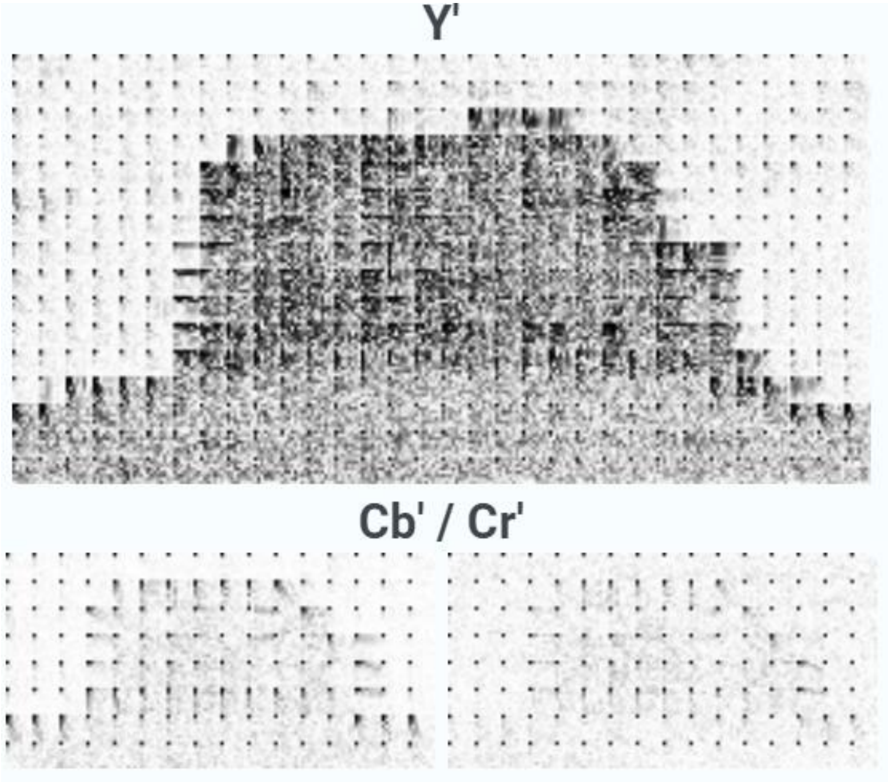
JGEG Encoding

- **Step 2: Reduce Color Information**
  - JPEG throws away some of the color information by **scaling down just the Cb & Cr (colour) channels** while **keeping the important *Y* (*brightness*) channel full size**.
  - The standard says you can keep all of the colour information, half of it, or a quarter of it.
    - *For images, most apps will keep half of the colour information; for video it is usually a quarter.*
  - **Notice**: 3 full channels and now we have 1 full channel and 2 × ¼ channels, for a total of 1½.



JGEG Encoding

- **Step 3: Conversion to Freq. Domain**
  - Start by dividing each of the Y, Cb, and Cr channels up into 8×8 blocks of pixels.
  - **Example**: *Consider just one of these 8×8 blocks from the Y channel.*
    - The spatial domain is what we have now: the value in the upper-left corner represents the brightness (Y-value) of the pixel in the upper-left corner of that block.
    - Domain transformation is accomplished using 2D Discrete Cosine Transform (DCT)



JGEG Encoding - DCT

- DCT : Discrete Cosine Transform
- DCT transforms the 64 values in 8x8 pixel block, in such a way that the relative relationship between the pixels are kept but the redundancies are revealed.
- Example.

A gradient grayscale

8 × 8

Block

20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90

P (x, y)

400	−146	0	−31	−1	3	−1	−8
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

T (m, n)

Quantization & Compression

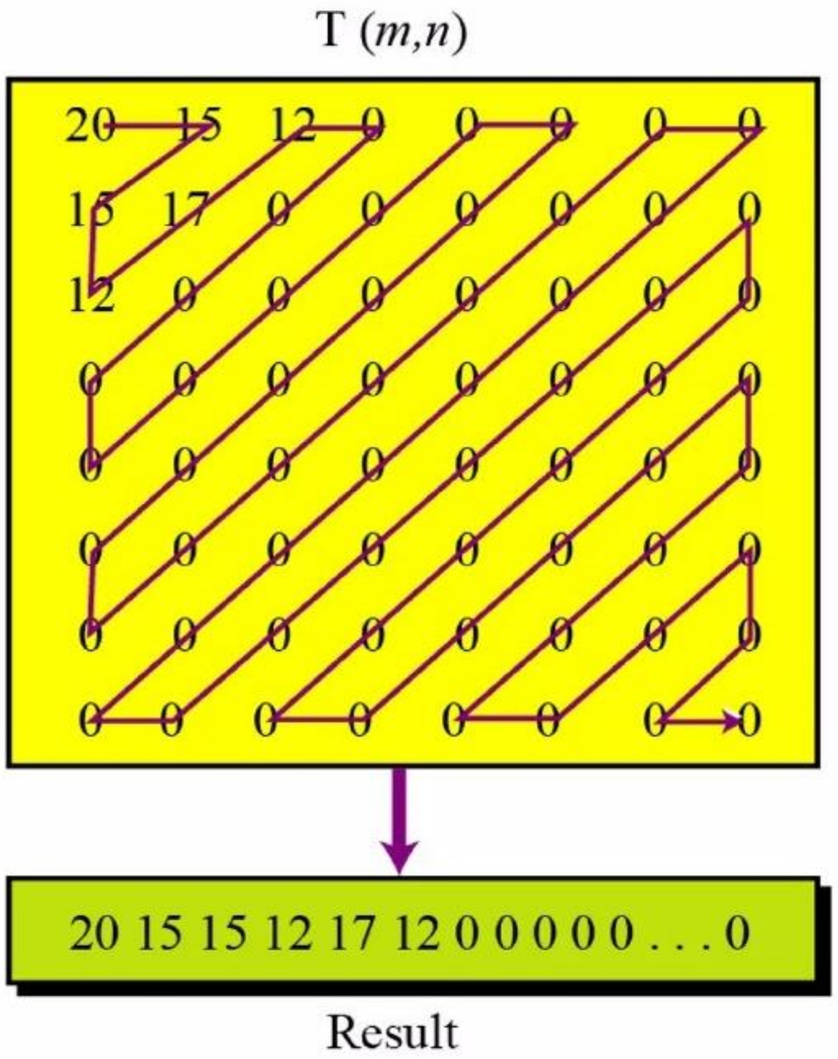
- **Quantization**
  - After Table T is created
  - The values are quantized to reduce the number of bits needed for encoding.
  - Quantization divides the number of bits by a constant.
  - Then drops the fraction.
  - Done to optimize the number of 0's for each particular application.

- **Compression**
  - Quantized vales are read from the table and the redundant 0's are removed.
  - To cluster the 0's together, the table is read diagonally, in a zigzag manner.
  - Reason: If the table doesnot have fine changes, the bottom right corner of the table is all 0's.
  - JPEG usually uses lossless run-length encoding at compression phase.

JPEG Encoding

JPEG has one last trick for making the data more compressible:

- It lists the values for each 8×8 block in a zig-zag pattern that **puts the numbers in order from lowest frequency to highest**.
- That means that the ***most heavily quantized parts*** (with the largest divisors) are ***next to each other*** to make nice, repetitive patterns of small numbers.



Block data is compressed in zig-zag order, grouping similar frequencies together.

MPEG Encoding

- Used for compressing video files.

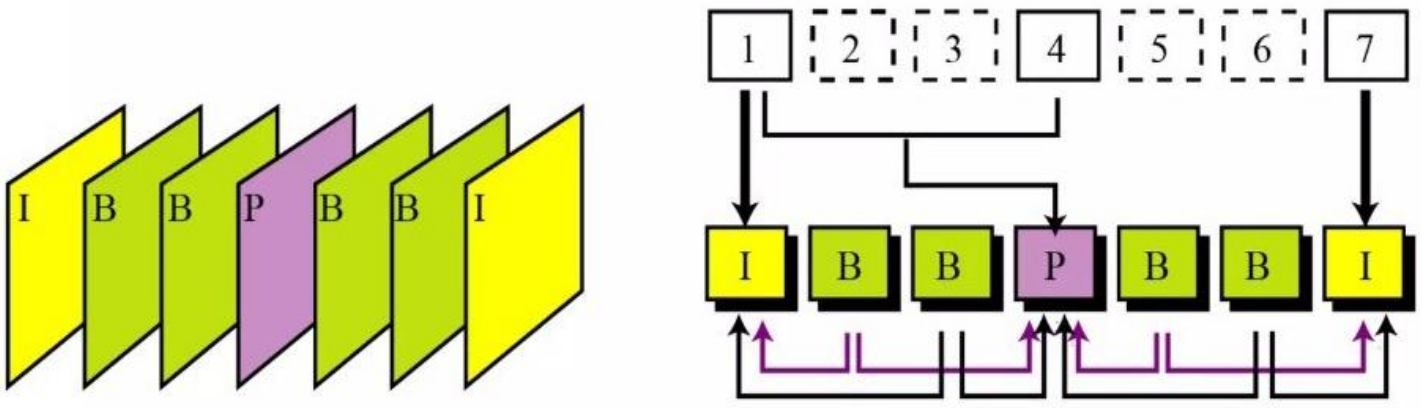
- **Basic Idea:**
  - Each video is a rapid sequence of a set of frames.
  - Each frame is spatial combination of pixels, or a picture.

- **Compressed Video** = Spatial Compression of Each Frame

  + Temporally compressing a set of frames

MPEG Encoding

- ***Spatial Compression***
  - Each frame is spatially compressed by JPEG
- ***Temporal Compression***
  - Redundant frames are removed
  - ***For Example:*** In a static scene in which someone is talking, most frames are the same expect for the sequence/ segment when the lips are moving, which change from one frame to other



a. Frames b. Frame construction

Audio Encoding

- Used for Speech/ Audio
  - Speech: compresses a 64 kHz digitized signal
  - Music: compresses a 1.411MHz
- Two Methods
  - Predictive Encoding
  - Perceptual Encoding

Audio Encoding

- **Predictive Encoding**
  - Only differences; samples are encoded not the whole sample values.
  - Several Standanrds: GSM (13 kbps), G.729 (8 kbps) & G723.3 (6.4 ot 5.3 kbps)

- **Perceptual Encoding MP3**
  - CD quality audio needs to be atleast at 1.411 Mbps
  - Cannot be sent over internt without compression
  - MP3 (MPEG Audio Layer 3) uses perceptual encoding to compress audio