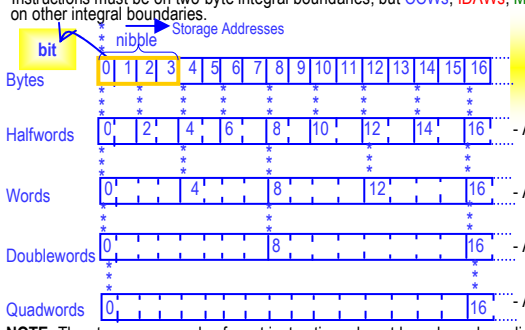
•Main storage provides the system with directly addressable fast-access storage of data.
•Both data and programs must be loaded into main storage (from input devices) before they can be processed.
•Main storage may include one or more smaller faster access buffer storages, sometimes called caches.
•A cache is usually physically associated with a CPU or an I/O (SAP) processor.
•The effects, except on performance, of the physical construction and use of distinct storage media are generally not observable by the program.
•Separate caches are maintained for instructions and for data operands where information within a cache is maintained in contiguous bytes on an integral boundary called a cache block or cache line (or line, for short).
•Fetching and storing of data by a CPU is not affected by any concurrent I/O (channel-subsystem) activity or by a concurrent reference to the same storage location by another CPU.
-When concurrent requests to a main-storage location occur, access normally is granted in a sequence determined by the system.
- If a reference changes the contents of the storage location, any subsequent storage fetches obtain the new content
•Main storage may be **volatile** or **nonvolatile**.
- If it is volatile, the contents of main storage are not preserved when power is turned off.
- If it is nonvolatile, turning power off and then back on does not affect the contents of main storage, provided all CPUs are in the stopped state and no references are made to main storage when power is being turned off.
NOTE: In both types of main storage, the contents of storage keys are not necessarily preserved when the power for main storage is turned off.

STORAGE ADDRESSING: Storage is viewed as a long horizontal string of *bits*.
•For most operations, accesses to storage proceeds in a left-to-right sequence.
-The string of bits is subdivided into units of eight bits.
- An eight-bit unit is called a *byte*, which is the basic building block of all information formats.
•Each byte location in storage is identified by a unique nonnegative integer, which is the address of that byte location or, simply, the byte address.
- Adjacent byte locations have consecutive addresses starting with 0 on left and proceeding in left to right manner.
- Addresses are unsigned binary integers and are **24, 31, or 64 bits**.
NOTE: X/A architecture debuted 31 bit addressing where the 32nd bit is used as a flag to denote 24 or 31bit mode.

INFORMATION FORMATS: Information is transmitted between storage and a CPU or the I/O channel subsystem one byte, or a group of bytes, at a time, unless otherwise specified, a group of bytes in storage is addressed bythe leftmost byte of the group.
- The number of bytes in the group is either implied or explicitly specified by the operation to be performed.
>When used in a CPU operation, a group of bytes is called a *field.*
•Within each group of bytes, bits are numbered in a left-to-right sequence. The leftmost bits are sometimes referred to as the "high- order" bits and the rightmost bits as the "low-order" bits.
-Bit numbers are not storage addresses, however, only bytes can be addressed. To operate on individual bits of a byte in storage, it is necessary to access the entire byte.
-The bits in a byte are numbered 0 through 7, from left to right.
•The bits in an address may be numbered 8-31 or 40-63 for 24-bit addresses or 1-31 or 33-63 for 31-bit addresses; they are numbered 0-63 for 64-bit addresses.
- Within any other fixed-length format of multiple bytes, the bits making up the format are consecutively numbered starting from 0.
NOTE: For purposes of error detection, and in some 'z' models for correction, one or more check bits may be transmitted with each byte or with a group of bytes where such check bits are generated automatically by the machine.
Parity checking is used where appropriate.

Integral Boundaries: Certain units of information must be on an integral boundary in storage and is called integral for a unit of information when its storage address is a multiple of the length of the unit in bytes.
•Special names are given to fields of 2, 4, 8, and 16 bytes on an integral boundary as illustrated below.
•When storage addresses designate halfwords, words, doublewords, and quadwords, the binary representation of the address contains one, two, three, or four rightmost zero bits, respectively.
•Instructions must be on two-byte integral boundaries, but CCWs, IDAWs, MIDAWs, and the storage operands of certain instructions must be on other integral boundaries.



A **nibble** is a four-bit aggregation or half an octet. .A full byte (octet) is represented by two hex digits; It is common to display a byte of information as two nibbles. The nibble is used to describe the amount of memory used to store a digit of a number stored in packed decimal format within a mainframe. This technique is used to reduce space requirements, make computations faster, and make debugging easier. An 8-bit byte is split in half and each nibble is used to store one digit.

- A halfword is a group of two consecutive bytes on a two-byte boundary and is the basic building block of instructions.
- A word is a group of four consecutive bytes on a four-byte boundary.
- A doubleword is a group of eight consecutive bytes on an eight-byte boundary.
- A quadword is a group of 16 consecutive bytes on a 16-byte boundary

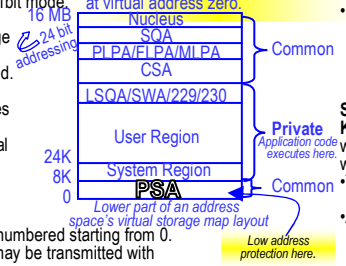
NOTE: The storage operands of most instructions do not have boundary-alignment requirements.
Address Types: For purposes of addressing main storage, **three 'basic' types of addresses** are recognized: ***absolute, real, and virtual***.
•The addresses are distinguished on the basis of the transformations that are applied to the address during a storage access.
•Address translation converts virtual to real, and prefixing converts real to absolute.

The **Channel-Command Word (CCW)** specifies the command to be executed and, for commands initiating certain I/O operations, it designates the storage area associated with the operation, the action to be taken whenever transfer to or from the area is completed, and other options.**CCW** indirect data addressing permits a single channel-command word to control the transfer of data that spans non-contiguous pages in real main storage. The use of CCW indirect data addressing also allows the program to designate data addresses above 16M bytes (format-0 CCWs) or above 2G-bytes (format-1 CCWs).

The Indirect-Data-Addressing-Word (IDAW) facility is provided to aid I/O operations in a virtual-storage environment. 64-bit IDAWs provide the only means by which data can be transferred directly between an I/O device and storage locations with addresses greater than 2G bytes.

The **Modified-CCW-Indirect-Data-Addressing (MIDA)** facility provides the program an alternate means to transfer large amounts of data that spans noncontiguous blocks in main storage without the overhead of data chaining and without the strict boundary and count restrictions imposed by the CCW.

Prefixing provides the ability to assign the range of real addresses 0-8191 to a different block in absolute storage for each CPU, thus permitting more than one CPU sharing main storage to operate concurrently with a minimum of interference, especially in the processing of interruptions. Prefixing causes real addresses in the range 0-8191 to correspond one-for-one to the block of **8K-byte** absolute addresses (the prefix). The Prefix Storage Area (**PSA**) starts at virtual address zero.



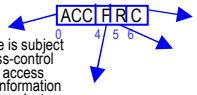
Absolute address is the address assigned to a main-storage location.
- An absolute address is used for a storage access without any transformations performed on it.
•The I/O channel subsystem and all CPUs in the configuration refer to a shared main-storage location by using the same absolute address.
- Available main storage is usually assigned contiguous absolute addresses starting at 0 and the addresses are always assigned in complete 4K-byte blocks on integral boundaries (unless specified with Large Page size - LFAREA).
- An exception is recognized when an attempt is made to use an absolute address in a block which has not been assigned to physical locations.
NOTE: On some models, storage-reconfiguration controls may be provided which permit the operator to change the correspondence between absolute addresses and physical locations. However, at any one time, a physical location is not associated with more than one absolute address.
•Storage consisting of byte locations sequenced according to their absolute addresses is referred to as absolute storage.

Real Address identifies a location in real storage.
•When a real address is used for an access to main storage, it is converted, by means of prefixing, to an absolute address.
- At any instant there is one real-address to absolute address mapping for each CPU in the configuration.
-When a real address is used by a CPU to access main storage, it is converted to an absolute address by prefixing.
- The particular transformation is defined by the value in the prefix register for the CPU.
- Storage consisting of byte locations sequenced according to their real addresses is referred to as real storage.

Virtual Address identifies a location in virtual storage.
•When a virtual address is used for an access to main storage, it is translated by means of dynamic address translation, (DAT) either (a) to a real address which is then further converted by prefixing to an absolute address, or (b) directly to an absolute address.

OTHER ADDRESS TYPES Except where otherwise specified, the storage-operand addresses for most instructions are ***logical addresses*** and are treated as real addresses in the real mode, as primary virtual addresses in the primary-space mode, as secondary virtual addresses in the secondary-space mode, as AR-specified virtual addresses in the access-register mode, and as home virtual addresses in the homespace mode (*using Cross Memory Mode– XMS*).
NOTE: Some instructions have storage-operand addresses or storage accesses associated with the instruction which do not follow the rules for logical addresses.
• Addresses used to fetch instructions from storage are called ***instruction addresses*** which are treated as real addresses in the real mode, as primary virtual addresses in the primary-space mode, secondary-space mode, or access-register mode, and as home virtual addresses in the home-space mode (*using Cross Memory Mode – XMS*). An ***effective address*** is the address which exists before any transformation by dynamic address translation or prefixing is performed. An effective address may be specified directly in a register (base address that serves as a displacement reference to other addresses) or may result from address arithmetic.

STORAGE PROTECTION – QoS: Prevents unauthorized access to storage owned by concurrent runtimes.
Key-Controlled Protection When key-controlled protection applies to a storage access, a store is permitted only when the storage key matches the access key associated with the request for storage access; a fetch is permitted when the keys match or when the fetch-protection bit of the storage key is zero.
•The keys are said to match when the **four access control bits** of the storage key are equal to the access key, or when the access key is zero.
•A **storage key** is associated with each 4K-byte block of storage that is available in the configuration and has the following format:



Access-Control Bits (ACC): If a reference is subject to key-controlled protection, the four access-control bits, bits 0-3, are matched with the four-bit access key when information is stored and when information is fetched from a location that is protected against fetching.

Fetch-Protection Bit (F): If a reference is subject to key-controlled protection, the fetch-protection bit, bit 4, controls whether key-controlled protection applies to fetch-type references: a zero indicates that only store-type references are monitored and that fetching with any access key is permitted; a one indicates that key-controlled protection applies to both fetching and storing. No distinction is made between the fetching of instructions and of operands.

Change Bit (C): The change bit, bit 6, is set to one each time information is stored at a location in the corresponding storage block.

Reference Bit (R): The reference bit, bit 5, Normally is set to one each time a location in the corresponding storage block is referred to either for storing or for fetching of information.

Storage Protection Summary

Conditions		Is Access to Storage Permitted	
Fetch-Protect Bit of Storage Key	Key Relation	Fetch	Store
0	Match	Yes	Yes
0	Mismatch	Yes	No
1	Match	Yes	Yes
1	Mismatch	No	No

Explanation:
Match - The four access-control bits of storage key are equal to the access key, or the access key is zero.
Yes - Access is permitted
No - Access is not permitted. On fetching, the information is not made available to the program; on storing the contents of the storage location are not changed.

Protection Four protection facilities are provided to protect the contents of main storage from destruction or misuse by programs that contain errors or are unauthorized: key-controlled protection, access-list-controlled protection, DAT protection, and low-address protection.
NOTE: The protection facilities are applied independently; access to main storage is only permitted when none of the facilities prohibits the access.
•Key-controlled protection affords protection against improper storing or against both improper storing and fetching, but not against improper fetching alone.
- When a CPU access is prohibited because of key controlled protection, the execution of the instruction is terminated, and a program interruption for a protection exception takes place.
•Access-register mode, bit 6 of the access-list entry, the fetch-only bit, controls which types of operand references are permitted to the address space specified by the access-list entry (used in Cross Memory Operations).
•DAT-protection function controls access to virtual storage by using the DAT-protection bit in each page-table entry and segment-table entry (not illustrated in this issue, but mentioned for completeness), and when the enhanced-DAT facility is installed, in each Region Table entry. It provides protection against improper storing.
•The low-address-protection facility provides protection against the destruction of main-storage information used by the CPU during interruption processing.
- The range criterion is applied before address transformation, if any, of the address by dynamic address translation or by prefixing.