on keyboard **10** by a user, critical data stored in memory **34**, a data stream transmitted to or from external computer **44**, or any executable code associated with memory **34**.

[**0054**] At event **120**, DMA controller **36** is set up for transferring a data stream to CRC circuit **38** from memory **34**. This setup will generally comprise providing a source address and a destination address for the data transfer, and the number of bytes involved in the data transfer, to DMA controller **36**.

[**0055**] At event **130**, the transfer of the data stream is initiated or started by DMA controller **36**, and the first byte of the data stream is transferred by DMA controller **36** to CRC circuit. DMA controller **36** may temporarily "seize" address and data transfer bus **30** to create a DMA channel for rapid transfer of the data stream to CRC circuit **38**.

[**0056**] At event **140**, DMA controller **36** continues to send bytes of the data stream to CRC circuit **38** via address and data bus **30** in accordance with the setup information provided to DMA controller **36** in event **120**.

[**0057**] At event **150**, DMA controller **36** makes a query as to whether all bytes in the data stream have been transferred to CRC circuit **38**. This query is made periodically after transfer of each byte. If all bytes in the data stream, as determined from the data stream size in the setup information from event **120**, have not been sent, then event **140** is repeated. If all bytes in the data stream have been transferred, then event **160** is carried out.

[**0058**] At event **160** a CRC value or check number is calculated for the data stream by CRC circuit **38** using a division algorithm.

[**0059**] At event **170**, the calculated CRC value is read from CRC controller circuit **38** into memory **34** for use in integrity checks for the data stream. At event **180**, the check value calculation is completed.

[**0060**] The invention also uses DMA controller **36** for fast transfer of data streams to display controller **40** in order to reduce software overhead and speed up data display. Prior art data processing systems have typically used programming to execute a loop to increment data, byte-by-byte, for output to a display controller, with the overhead of the software loop increased by each byte written to the display driver. The invention overcomes this deficiency by carrying out transfer of data directly from memory **34** to display controller **40** via DMA controller **36**, with programming used only in the setup or configuration of the DMA controller **36** and display controller **40**, and to initiate the data transfer by DMA controller **36**. By eliminating the software overhead involved in transferring bytes from memory to the display controller **40**, the transfer is performed rapidly, and the display of information (text and/or graphics) takes less time, which is appreciated by the user viewing the display **22**. Use of the DMA controller **36** to drive display controller **40** in this manner can provide timesavings of up to 80% or more over the conventional use of a software loop to write pixel data bytes to a display controller.

[**0061**] Data display using DMA controller **36** in accordance with the invention will be more fully understood by reference to **FIG. 5**, as well as **FIG. 3**. At event **200**, software **50** loaded in memory **34** is started or initiated which includes programming for carrying out operations associated with setup of display controller **40**, setup of DMA controller **36**, and initiation of the transfer of data to display controller **40** by DMA controller **36**.

[**0062**] At event **210**, display controller **40** is set up by providing display controller **40** with a "write" command and a display destination address (in the memory of display controller **40**) for the data stream to be transferred to display controller **40**. The data stream may comprise any dipslayable data such as, for example, graphical user interface (GUI) data associated with stored programming for display of icons or other features, data associated with a character or string of alphanumeric characters entered on keyboard **10** by a user, or other data stored in memory **34**.

[**0063**] DMA controller **36** is set up in event **220** for transferring a data stream to display controller **40** from memory **34**. This setup comprises providing a memory source address and the display destination address for the data transfer, and the number of bytes involved in the data transfer, to DMA controller **36**.

[**0064**] At event **230**, the transfer of the data stream is initiated or started by DMA controller **36**, and the first byte of the data stream is transferred by DMA controller **36** to display controller **40**. DMA controller **36** may temporarily "seize" address and data transfer bus **30** as described above to create a DMA channel for rapid transfer of the data stream to display controller **40**.

[**0065**] At event **240**, DMA controller **36** continues data transfer by sending the next byte of the data stream to CRC circuit **38** via address and data bus **30**.

[**0066**] At event **250**, DMA controller **36** makes a query as to whether all bytes in the data stream have been transferred to display controller **40** according to the setup information provided to DMA controller **36** in event **220**. This query is made periodically after transfer of each byte. If all bytes in the data stream, as determined from the data stream size in the setup information from event **220**, have not been sent to display controller **40**, then event **240** is repeated. If all bytes in the data stream have been transferred, the process is completed at event **260**.

[**0067**] The invention also provides for increased efficiency in data processing devices by decreasing the time and memory requirements needed for validation of user identification (ID) character strings. Hand held data processor devices typically do not have sufficient memory to store an uncompressed ID character strings. In this regard, the invention provides for compression of user ID strings by creation of CRC or check values for each authorized ID string.

[**0068**] The CRC values for the authorized strings are sorted and stored in a list or lookup table in the memory of an external computer **44**. External computer **44** may comprise, for example, a minicomputer, a microcomputer, a UNIX® machine, a mainframe, a personal computer (PC) such as an INTEL® based processing computer or clone thereof, an APPLE® computer or clone thereof, or a SUN® workstation, or other appropriate computer with conventional hardware components (not shown) such as a motherboard, central processing unit (CPU), random access memory (RAM), hard disk drive, display adapter, other storage media, a monitor, keyboard, mouse, and other user interface means, a network interface card (NIC), floppy disk drive, CD drive, and/or other conventional input/output devices.