DIGITAL TIME CODE SYSTEM

Matthew R. Flax
School of Electrical Engineering & Telecommunications,
University of NSW
flatmax@ieee.org
Jesse S. Jin
School of Computer Science & Engineering, University of NSW
Department of Computer Science, University of Sydney
jesse@cs.usyd.edu.au

---

proceedings - cybersonica 2002, pages 61-64
http://www.cybersonica.org/

DIGITAL TIME CODE SYSTEM

Matthew R. Flax
School of Electrical Engineering & Telecommunications, University of NSW
flatmax@ieee.org
Jesse S. Jin
School of Computer Science & Engineering, University of NSW
Department of Computer Science, University of Sydney
jesse@cs.usyd.edu.au

***Abstract*— A time code system is described which allows linear and non-linear alteration for multimedia systems and streams. The time code system is adaptable to any time code protocol. The time code system is computationally inexpensive.**

I. INTRODUCTION

Time code is temporal control of multimedia systems and streams. Examples of systems which are controlled by time code are ATA packet interface (ATAPI) devices [1] such as CDROM drives, music instrument digital interface (MIDI) devices [2] such as hardware MIDI sequencers and much video SMPTE [3] based hardware. Streams and systems alike are controlled by many specific time code protocols such as SMPTE, midi time code (MTC), ATAPI MSF address and so on.

The specific aim of this article is to outline a time code abstract data type which is suitable for human computer interaction and bears low computational complexity. By the same token such a time code must be associated with a window of media samples or frames. The time code interfaces a window of media to a large stream of media windows, which may overlap. The data type which is abstracted is typically a natural number, termed an unsigned integer.

The abstract data type is applicable to ANY time code protocol. It is not a protocol. It is an abstraction of a natural number to a user specified protocol. Each time code reference is allocated a media data buffer. Time and data are closely coupled. These two attributes are unique to this time code abstract data type.

A. HCI

Alteration of time code allows the user to position oneself non-linearly in a multimedia device or stream. The only element which limits the user to accurate and direct location in a stream should be channel bandwidth, where the user may specify a location in time, however bandwidth congestion in the media transport limit the direct location. Linear and non-linear interface to time code should be possible. Linear interface typically compromises accuracy in location for ease of location. Non-linear interface allows accuracy however compromises ease of location.

An example of a linear time code locator is the location slider. Using this type of device it is easy to locate oneself non-linearly in a stream, however location is limited to the slider accuracy, which is typically quantised. As an example of quantisation, assume a horizontal slider is used to locate an audio stream which is of compact disk (CD) quality and is of one hour in length. Also assume the slider spans one thousand pixels and hence has one thousand possible locations. CD quality audio has 44100 frames per second. An hours worth of audio contains around $159x10^6$ frames. Each slider location quanta is $159x10^3$ frames of audio, that is 3.6 seconds of audio. In this example media location accuracy is compromised by quantisation to within 3.6 seconds of the desired location.

An example of non-linear time code location is digit based alteration. An example of such a location alteration method is the setting method used by digital watches based on digit interface. To locate oneself in time one has to step through time digit by digit. It takes more effort to locate oneself in time, however accuracy is not compromised.

B. Computational Complexity

Integer manipulation is the cheapest data type for computer processors to manipulate. In order to retain cheap computational complexity, all operations on time code are implemented in integer operations of addition, subtraction, multiplication, division and modulus. By the same token it is desired that digit based abstraction of the integer operates with the same minimal computational complexity. For this reason it is possible to implement such an abstraction system using object inheritance.

II. TIME CODE ABSTRACTION

Label the base integer a *counter*, this is its job, to count. It is the base most object.

Assume that time code is broken into many *field*s. As an example world time is broken into many many fields, to name a few from largest time to least {millennium, centuries, decades, years, months, weeks, days, hours, minutes, seconds, milliseconds}. Each field must retain its count and inherits the counter object to accomplish this.

A master is required to manage the co-ordination of each of the fields of the time code with respect to the underlying base counter. At each point in time the individual fields must accurately represent (count up to) the same total which is counted by the *master counter* element. The master counter element maintains the master count by inheriting the base counter object.

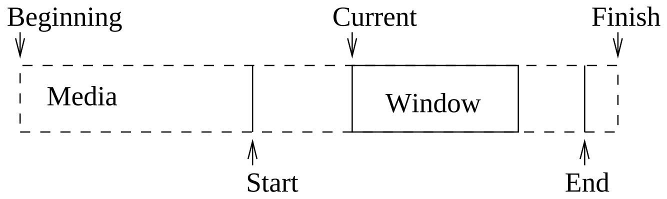

Fig. 1. Time code structure. Physical media limits are specified by the 'Beginning' and 'Finish'. A desired media segment is specified by the 'Start' and 'End'. The current location of the media window is specified by the 'Current' counter.

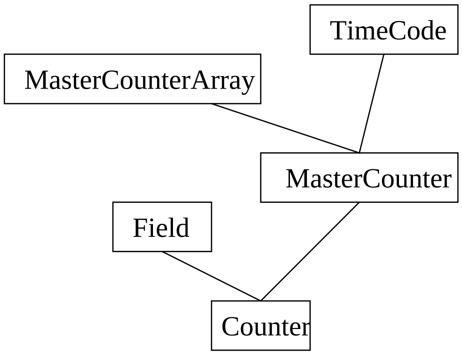

Fig. 2. Time code object inheritance hierarchy.

A window or array of frames is also based on a master counter. The user might want to alter the window size in a linear or non-linear way. So each media window or array of frames must be controlled by a master counter, labelled a *master counter array*. A master counter is inherited for array or window size manipulation.

Finally *time code* is encapsulated by a beginning and a finish. These are the absolute limits of the media stream beyond which no media is defined or exists. Within the physical limits one may set a start and end location between which a window steps. A current time code location continuously steps with the window. Such a structure seems complex, however entails the requirements of a manageable and flexibly alterable time code system. This system is depicted in Figure 1.

Figure 2 depicts the specified inheritance hierarchy.

Internal to each object in the time code hierarchy are elements which are required for time code system operation. Figure 3 depicts the hierarchy and its internal elements. Data types specified are in the C programming language notation and are as follows :

int	An integer number limited by word size.
uint	An unsigned integer which is greater or equal to zero.
uchar	An unsigned eight bit number which is greater then or equal to zero.
TYPE *	A memory pointer to an array of TYPE.
TYPE **	An array of memory pointers to arrays of TYPE.

Counter, the base of the hierarchy (located at the bottom of the Figure). The counter counts between a minimum and maximum value. If any operations force the count above or below the max/min, then carry is indicated. This is essentially a self contained arithmetic logic unit ALU. Inherent in this is a looping mechanism, where if the maximum is exceeded, then a modulus operator loops to the remainder above the minimum. The same applies for the case when the minimum is under-ceded.

Field, a leaf node in the hierarchy (located middle left in the Figure). The field is a counter with a digit display. The counter maximum determines the number of digits required by the field, called the digit count. Each digit is alterable in a field, either incremented or decremented. When a digit is altered, the inherited counter adjusts accordingly such that the counter represents the number indicated by the digits. The counter is also alterable, in which case the digits adjust to represent the counter value. As the number of digits depends on the maximum value, digits are dynamically created and pointed to in computer memory space.

Master Counter, a limb in the hierarchy (located middle right in the Figure). The master counter maintains the master count. It synchronises the individual fields with the master count. When the master count is altered, then relevant fields are altered to add in total at all times to the master count. Again vice versa applies, when a field is altered, the master count is constantly the sum of all the fields. A master count may be split into arbitrary numbers of fields. Each field is dynamically created and pointed to in memory space. At this point in the hierarchy, the tasks of linear and non-linear representation are accomplished. Linearly the base counter object is adjustable. Non-linearly individual digits in individual fields are adjustable.

Master Counter Array, a leaf node in the hierarchy (located top left in the Figure). This object controls the size of the window (a dynamically allocated array) of media frames. A master counter allows alteration of the size of the media window or frame count. Frame size is specified in terms of word count. The data type of the array is adjustable.

Time Code, a leaf node in the hierarchy (located top right). This object encapsulates time code which is consistently within the physical limits of the underlying media stream or system. Start and end points, as well as window size are flexibly alterable. Such a time code (t) is made of a beginning (b), a start (s), a current (c), an end (e), a finish (f) and a window (w). At all times $b \le s \le e \le f$. This allows the start and end to shift but not overlap. The current location obeys $s \le c < e$. The window should have the following properties $w \le (e - s)$. This defines a multimedia time code.

III. EXPERIMENT

An implementation of the time code hierarchy is available [4]. This is used in experiment to give examples of each element in operation.

Counter, in this example three initial counters $\{c, d, e\}$ are set up where $\left\{ \begin{array}{ll} c = 212; & 0 \le c < 300 \\ d = 13; & 0 \le d < 300 \\ e = 0; & 0 \le e < 100 \end{array} \right\}$. The following is computed $e = e + (c + d)$ this yields the result

$$e = 25; \; carry = 2$$

an interpretation of this result is that the addition $(c + d)$ yields 225, when added to e yields 225, which wraps twice above the maximum value of e. The result is the modulus $e = 25$ and two carries.

Field, in this example a field (f) is set up such that $f = 1;\ 0 \le f < 21$. As the field maximum is set to twenty one,

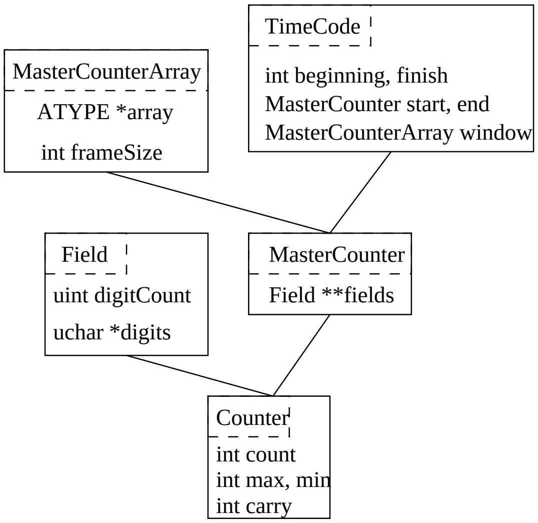

Fig. 3. Time Code hierarchy internal elements

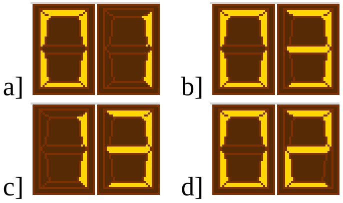

Fig. 4. Field example. a] Initial value, b] unit digit increment twice, c] tens digit increment, d] tens digit increment

two digits are required to represent this field. Three operations are carried out on this field and each of the results are represented in Figure 4. The initial value of the field is one. Operation 'b]' increments the unit digit twice, hence adding two to the underlying counter. The value of the field is now $f = 3$. Operation 'c]' increments the tens digit. The value of the field is now $f = 13$. Operation 'c]' increments the tens digit again. In this case the counter carries once as its value is $f = 23$, as the maximum is exceeded the field and counter now equals two, $f = 2$; $carry = 1$.

Master Counter, we create a master counter (c) with three fields, hours (h), minutes (m) and seconds (s) in this case we require the following conditions to apply

$$\begin{array}{ll} h = 0; & 0 \leq h < 24 \\ m = 0; & 0 \leq m < 60 \\ s = 0; & 0 \leq s < 60 \end{array}$$

We also choose a minimum of two minutes and a maximum of twenty three hours, fifty nine minutes and fifty nine seconds is assumed. Hence in seconds, $120 \leq c < 86400$ and initially $c = 120$. A single operation is carried out on the master counter. Figure 5 depicts these operations. Initially c is set to two minutes (one hundred and twenty seconds) as that is the minimum possible value. Operation 'b]' decrements the seconds unit digit by one. As the minimum value is under-ceded, the master counter wraps to the highest value, namely twenty three hours, fifty nine minutes and fifty nine seconds. The carry is set to negative one indicating under-ceding the minimum once.

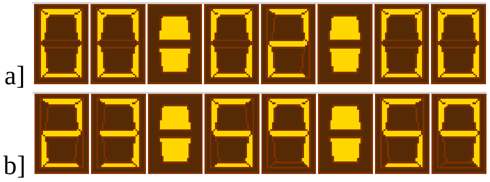

Fig. 5. Master counter example. a] Initial value, b] Seconds unit digit decrement.

IV. CONCLUSION

Linear time code representation allows mathematical manipulation. Simple media location is also accomplished, however accuracy is compromised. Non-linear time code representation allows accurate media frame location. Such a high level description of the media stream location (field by field, digit by digit) is good for interaction.

This article and the software referenced is unique in its ability to control any time code protocol. Media data is closely coupled to the time reference and this is an advantage for systems which require media data shifting and location.

REFERENCES

[1] Small Form Factor Committee Specification of ATA Packet Interface for CD-ROMs. SFF-8020i

[2] Complete MIDI 1.0 Detailed Specification, MIDI Manufacturers Association

[3] Society of Motion Picture and Television Engineers (SMPTE) time code

[4] Time code implementation available from : http://mffmtimecode.sourceforge.net/