(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 March 2007 (01.03.2007)

PCT

(10) International Publication Number
WO 2007/024543 A2

(51) International Patent Classification:
G06K 17/00 (2006.01) *G06K 7/10* (2006.01)

(21) International Application Number:
PCT/US2006/031671

(22) International Filing Date: 14 August 2006 (14.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/709,433 19 August 2005 (19.08.2005) US

(71) Applicant *(for all designated States except US)*: **GLOBAL PAYMENT TECHNOLOGIES, INC.** [US/US]; 425b Oser Avenue, Hauppauge, NY 11788 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **MAZOWIESKY, Thomas** [US/US]; 28 Clifton Drive, Patchogue, NY 11772 (US).

(74) Agent: **GARRETT, Arthur, S.**; Finnegan, Henderson, Farabow, Garrett & Dunner LLP, 901 New York Avenue, NW, Washington, DC 20001 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2007/024543 A2

(54) **Title:** INFORMATION READERS, APPARATUSES INCLUDING INFORMATION READERS, AND RELATED METHODS

(57) **Abstract:** This invention relates to the field of information reading, and more specifically, to an improved information reader in a validator which eliminates the requirement to insert an information source in a particular direction in an information reader.

INFORMATION READERS, APPARATUSES INCLUDING INFORMATION READERS, AND RELATED METHODS

DESCRIPTION OF THE INVENTION

This application claims priority to U.S. Provisional Patent Application No. 60/709,433, filed on August 19, 2005.

Field of the Invention

[001] This invention relates to the field of barcode reading, and more specifically, to an improved barcode reader in a currency validator which eliminates the requirement to insert a barcode coupon in a particular direction in a reader.

Background of the Invention

[002] In the casino gaming industry, the application of barcode coupons has risen in recent years. These 'cashless' systems allow a player to transfer funds using a printed barcode coupon.

[003] The person playing the game inserts a coupon into the slot game, via the currency validator. The currency validator reads the coupon and transmits the coupon number to the game. The game then transmits the coupon number to a central system, which verifies the coupon number, and then transmits a stored linked value to the game. The game credits the customer with the transmitted value, allowing the customer to play the game. After the customer completes game play, he 'cashes out' by hitting a button on the game. The game signals the central system, transferring the credit to the system. The system then instructs the game to print a new barcode coupon. The user can then take the coupon and play another game, or go to the cashier and receive money in exchange for the coupon.

[004] This system allows the casino to eliminate much of the time and expense devoted to cash handling. Cash need not be secured, counted, guarded, etc – the only effort expended is for a computer system to keep track of the value the customer has earned while transferring between games. Since the coupons have no inherent value once redeemed by the system, used coupons are discarded and need not be handled, as currency would be.

[005] Until now, the validators that read the coupons have been limited to reading coupons either face up or face down when inserted in the reader, as the validators contain only one barcode reader device either above or below the bill channel. When inserted in the wrong direction, the coupon will be rejected, causing the customer to re-insert the coupon until he hopefully discovers his error and inserts it properly. This limitation can confuse the customer and cause play to be delayed, thus reducing the revenue to the casino. In some cases, the customer may get frustrated to the point that he cashes out and takes his business elsewhere.

SUMMARY OF THE INVENTION

[006] One object of the disclosed invention may be to eliminate confusion and delay on the part of a customer by placing a barcode reading device both above and below a coupon. The barcode reading device may simultaneously read both sides of the coupon. The results from the reader may be used to transfer a number from the coupon to a host computer system

[007] It may be a further object of the invention to allow a coupon to be printed with barcodes on both sides of the coupon, and then read by the dual barcode reader. This may permit a much larger coupon number to be used in the barcode system. Systems today are limited to about 20 characters.

[008] An embodiment of the invention may include a device for validating barcoded coupons. The device may include a means for forming a bill channel with an entrance and exit, a means for positioning a barcode reading detector above the bill channel, a means for positioning a barcode reading detector below the bill channel, a means to transport the coupon so that the barcode symbols on the coupon are moved perpendicular to the barcode detectors, a means to sample the signal from the barcode detectors, a means for storing digital representations of the samples from the barcode detectors, and/or a means for analyzing the digital representations of the sampled signals to determine and decode the coupon number and/or the digital representations stored therein.

[009] Another embodiment of the invention may include a device for validating barcoded coupons. The device may include a means for forming a bill channel with an entrance and exit, a means for positioning a barcode reading

detector above the bill channel, a means for positioning a barcode reading detector below the bill channel, a means to transport the coupon so that the barcode symbols on the coupon are moved perpendicular to the barcode detectors, a means to sample the signal from the barcode detectors constructing a digital representation of the sampled signal, and/or a means for analyzing the digital representations of the sampled signals to determine and decode the coupon number and/or the digital representations stored therein.

[010] Various embodiments of the invention may include one or more of the following aspects: the number decoded from the barcode detectors may be combined together to form a decoded coupon number with the sum of digits of the separate barcode numbers; the number decoded may be the product of the separate barcode coupon numbers; and the separate coupon numbers may be prime numbers.

[011] A further embodiment of the invention may include a validator. The validator may include at least two information readers configured to read information from an information source and a transporter configured to physically transport the information source past the at least two information readers such that at least one of the at least two information readers reads information from the information source.

[012] Various embodiments of the invention may include one or more of the following aspects: the at least two information readers may be configured to read information from different sides of the information source; the at least two information readers may be offset from each other in a direction of transport of the information source; a housing; the at least two information readers may be disposed in the housing such that each of the at least two information readers can only read information disposed on one side of the information source; the at least two information readers may be disposed in the housing such that a first of the at least two information readers can only read information disposed on a first side of the information source and a first of the at least two information readers can only read information disposed on a second side of the information source different from the first side; and a system including the validator.

[013] Yet another embodiment of the invention may include a validator. The validator may include a first information reader configured to read a first information portion disposed on a first portion of an information source, a second

information reader configured to read a second information portion disposed on a second portion of the information source different from the first portion, a transporter configured to physically transport the information source past the first information reader and the second information reader such that the first information reader reads the first information portion from the first portion of the information source and the second information reader reads the second information portion from the second portion of the information source, and a microprocessor configured to combine the first information portion and a second information portion into a third information portion.

[014] Various embodiments of the invention may include one or more of the following aspects: a housing; the first information portion may be disposed in the housing on a first side of the information source and the second information portion is disposed in the housing on a second side of the information source different from the first side; the at least two information readers may be offset from each other in the housing in a direction of transport of the information source; the first information portion may be a first barcode portion; the second information portion may be a second barcode portion; and the third information portion may be a complete barcode; the complete barcode may have more than 20 characters; and a system including the validator.

[015] Still another embodiment of the invention may include a method. The method may include providing a validator including at least two information readers, inserting an information source into the validator, transporting the information source past the at least two information readers, and reading information from the information source via at least one of the at least two information readers.

[016] Various embodiments of the invention may include one or more of the following aspects: reading information from the information source via both of the at least two information readers; reading a first information portion via a first of the at least two information readers and reading a second information portion via a second of the at least two information readers; reading a first information portion from a first side of the information source via a first of the at least two information readers and reading a second information portion from a second side of the information source via a second of the at least two information readers; and a first information portion may be read by a first of the at least two information readers during a first time

interval and a second information portion may be read by a second of the at least information readers during a second time interval different from the first time interval.

[017] A still further embodiment of the invention may include a method. The method may include providing a validator including a first information reader and a second information reader, inserting an information source into the validator, transporting the information source past the first information reader and the second information reader, reading a first information portion from the information source via the first information reader and a second information portion from the information source via the second information reader, and combining the first information and the second information portion into a third information portion.

[018] Various embodiments of the invention may include one or more of the following aspects: the first information portion may be disposed on a first side of the information source and the second information portion may be disposed on a second side of the information source different from the first side; the first information portion may be read by the first information reader during a first time interval and the second information portion may be read by the second information reader during a second time interval different from the first time interval; the first information portion may be a first barcode portion, the second information portion may be a second barcode portion, and the third information portion may be a complete barcode; and the complete barcode may have more than 20 characters.

[019] Additional objects and advantages of the invention will be set forth in part in the description that follows, and in part will be obvious from the description, or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

[020] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

[021] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and together with the description, serve to explain the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[022] Fig. 1 depicts a validator, according to an exemplary embodiment of the invention.

[023] Fig. 2 depicts an information reader of the validator of Fig. 1.

[024] Figs. 3A and 3B depict an information source, according to another embodiment of the invention.

[025] Fig. 3C depicts an information source, according to yet another embodiment of the invention.

[026] Fig. 3D depicts an information source, according to a yet further embodiment of the invention.

[027] Fig. 4 depicts an apparatus including the validator of Fig. 1.

[028] Fig. 5 depicts a method of processing information, according to a further embodiment of the invention.

[029] Figs. 6A and 6B depict a method of reading information, for example, a barcode, according to still another embodiment of the invention.

DESCRIPTION OF THE EMBODIMENTS

[030] Reference will now be made in detail to the exemplary embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

[031] Fig. 1 depicts a validator 100 according to an embodiment of the present invention. Validator 100 may include a housing 1 and may be disposed in apparatus 300, for example, as shown in Fig. 4. Housing 1 may house, accommodate, and/or define one or more of insertion channel(s) 2, actuator(s) 3, gear(s) 4, 5, 6, pulley(s) 7, drive belts 8, 9, first information reader 10, second information reader 11, roller(s) 12, drive wheel(s) 14, first signal source 15, first signal detector 16, and microprocessor 17. Validator 100 may be disposed in any suitable apparatus 300 as shown in Fig. 4, for example, a slot machine.

[032] Portions of validator 100 defining and/or surrounding insertion channel 2 may be configured to receive and transport an information source 200, examples of which are shown in Figs. 3A-3D, inserted therein. Specifically, a

combination of one or more actuator(s) 3, gear(s) 4, 5, 6, pulley(s) 7, drive belts 8, 9, roller(s) 12, and drive wheel(s) 14 may be configured to receive and transport information source 200, for example, those shown in any one of Figs. 3A-3D.

[033] First signal source 15 and first signal detector 16 may be disposed inside housing 1 and/or around insertion channel 2 such that when no information source 200 is disposed in validator 100 first signal detector 16 receives a substantially continuous signal from first signal source 15. When information source 200 is inserted into insertion channel 2, however, information source 200, or some other portion of validator 100 configured to move due to the insertion of information source 200, may partially or completely block the signal from first light source 15 and prevent all or part of the signal from reaching first signal detector 16. First signal detector 16 may then inform microprocessor 17 that it is no longer receiving a signal from first signal source 15 (e.g., by sending a signal or ceasing to send a signal), and in turn microprocessor 17 may activate actuator 3. Actuator 3 may then cause roller(s) 12 and drive wheel(s) 14 to transport information source 200 via one or more of gear(s) 4, 5, 6, pulley(s) 7, and drive belts 8, 9. Information source 200 may be transported in any direction, for example, away from insertion channel 2. If information 201 is a barcode, validator 100 may be configured to move information 201 in a direction substantially perpendicular to information readers 10, 11. Microprocessor 17 may be configured to transport information source 200 through validator 100 at any rate, for example, at a substantially constant and/or steady rate such that information source 200 does not stop in insertion channel 2. This may be to assist information readers 10, 11 in reading information 201 from information source 200. It should be understood, however, that information source 200 may transport through validator 100 using any known device and/or method.

[034] As shown in Fig. 2, information readers 10, 11 may each include one or more of housing 18, second signal source 20, second signal detector 19, and slit 21. Information readers 10, 11 may include a barcode reader, for example, a barcode reader manufactured by OPTO TECHNOLOGY, INC. including the OTBC-06XX series (e.g., OTR 680/690). Information readers 10, 11 may operate using any known method and/or algorithm. For example, second signal source 20 may project a signal through slit 21. The signal may impinge on an information source, for example, information source 200. Information source 200 and/or information 201

disposed on information source 200 may alter and/or reflect at least a portion of the signal from second signal source 20 to second signal detector 19. Second signal detector 19 may detect the altered/reflected signal and/or information 201 and send it to microprocessor 17 for processing. Information readers 10, 11 may be configured such that unless second signal detector 19 is detecting a signal altered and/or reflected by information 201 (e.g., a barcode), second signal detector 19 will not send a signal and/or information 201 to microprocessor 17. In the alternative, information readers 10, 11 may send a continuous signal to microprocessor 17, and microprocessor 17 may determine which of these continuous signals is information 201.

[035] In a further example, information readers 10, 11 (or, depending on the process, one of microprocessors 17, 317, 417) may implement the following exemplary algorithm set forth in Figs. 6A and 6B.

[036] In another example, information readers 10, 11 (or, depending on the process, one of microprocessors 17, 317, 417) may implement the following exemplary algorithm set forth in C code:

```
#define _BARCODE_
#include   <string.h>
#include   "model.h"
#include   "processor.h"
#include   "..\inc\vconst.h"
#include   "..\inc\appdata.h"
#include   "..\recog\inc\datacoll.h"
#include   "..\app\inc\validate.h"
#include   "..\recog\inc\barcode.h"


#define WIDE_PULSE                  0
#define BC_LEFT_TO_RIGHT      1
#define BC_RIGHT_TO_LEFT      0
#define MAX_BC_DIGITS         24
#define MIN_BC_DIGITS         14
#define MAX_FLAT_POINTS             80
```

```
#define MAX_FALL_CNT                    (MAX_BC_DIGITS * 5)/2 + 4 /* 64
    */
#define MIN_FALL_CNT                    (MIN_BC_DIGITS * 5)/2 + 4 /* 39    */

static UCHAR    XDATA    Dp_Width[ MAX_FALL_CNT ] ;
static UCHAR    XDATA    Pk_Width[ MAX_FALL_CNT ] ;

static UCHAR    XDATA    Dp_Value[ MAX_FALL_CNT ] ;
static UCHAR    XDATA    Pk_Value[ MAX_FALL_CNT ] ;

static UWORD    XDATA    Dp_WxA[ 5 ] ;      /*    WxA = (Width * Ampl)
of bars - dips */
static UWORD    XDATA    Pk_WxA[ 5 ] ;      /*    WxA = (Width * Ampl)
of spaces - peaks */

static UCHAR    XDATA    Max_High ;    /*    highest peak value */
static UCHAR    XDATA    Min_Low ;     /*    lowest dip value   */

static UWORD    XDATA    fall_cnt, rise_cnt ;
static UWORD    XDATA    last_f_start_ix, last_f_end_ix ;
static UWORD    XDATA    prev_f_start_ix, prev_f_end_ix ;
static UWORD    XDATA    last_r_start_ix, last_r_end_ix ;
static UWORD    XDATA    prev_r_start_ix, prev_r_end_ix ;
static UWORD    XDATA    bc_data_samples;

static UCHAR    XDATA    tot_bars ;
static UCHAR    XDATA    dig_bars ;    /* bars representing data only, less
start and end */
static UCHAR    XDATA    bc_order ;
static UCHAR    XDATA    bc_error_flag ;

static UCHAR  XDATA*  bc_buffer_ptr ;
```

```
/* note location table pointer        */
static  void    **XDATA nltptr;

static UCHAR        FXDATA  output_buffer[ MAX_BC_DIGITS ] ;
static UCHAR        FXDATA  decodecharcount;

static void     Find_Dips_And_Peaks(void);
static UCHAR        Get_BC_Data_Order(void);
static void     Find_And_Decode2of5(void);
static UCHAR        Decode_Digit(UCHAR dig);

static UCHAR        Edge_IsFalling(UCHAR x0,UCHAR x1,UCHAR x2,UCHAR x3);
static UCHAR        Edge_IsRising( UCHAR x0,UCHAR x1,UCHAR x2,UCHAR x3);

static void     Get_Peak_Value( UWORD start_ix );
static void     Get_Dip_Value(  UWORD start_ix );
static void     Get_3_Greatest( UWORD* dmax_arr,UWORD* pmax_arr, UCHAR*
bar, UCHAR* spc );

static void     Clear_BC_Data(void);

/* returns a null term. str. with the coupon num.  max length is 24  chars */
UCHAR* gBarcode_GetCoupon(void)
{
   return(&output_buffer[0]);
}
/* this function returns the # of characters in the barcode coupon */
UCHAR gBarcode_GetCouponLength(void)
{
   return (decodecharcount);
}

static void Clear_BC_Data(void)
```

```
{
UCHAR i;

        last_f_start_ix     = 0;
        last_f_end_ix       = 0;
        prev_f_start_ix     = 0;
        prev_f_end_ix       = 0;
        last_r_start_ix     = 0;
        last_r_end_ix       = 0;
        prev_r_start_ix     = 0;
        prev_r_end_ix       = 0;
        Min_Low             = 250;
        Max_High            = 0;
        fall_cnt            = 0;
        rise_cnt            = 0 ;
        tot_bars            = 0;

        for( i = 0; i < MAX_FALL_CNT; i++)
        {
                Dp_Width[ i ] = 0 ;
        Pk_Width[ i ] = 0 ;
        Dp_Value[ i ] = 0;
        Pk_Value[ i ] = 0;
   }

        for( i = 0; i < 5; i++)
        {
                Dp_WxA[ i ] = 0 ;
                Pk_WxA[ i ] = 0 ;
        }
}

/*      If the coupon's entry point was the barcode's logical start, */
```

```
/*      returns BC_LEFT_TO_RIGHT, otherwise  BC_RIGHT_TO_LEFT          */

static UCHAR Get_BC_Data_Order(void)
{
UCHAR  max_w, min_w, w;
UCHAR  i,ret;

        // if( bc_error_flag > 0) return 0;

        ret = BC_LEFT_TO_RIGHT;

        max_w       = 0;
        min_w= 250;
        for( i = 2; i < 7; i++)
        {
                w = Dp_Width[i];
                if( w <= min_w )  min_w = w;
                if( w >= max_w )  max_w = w;
        }
        if(( max_w - min_w ) >= 22 )
        {
                if( Dp_Width[1] > Dp_Width[ tot_bars - 2] )
                        ret = BC_RIGHT_TO_LEFT;
        }
        else
        {
                if( Dp_Value[1] < Dp_Value[ tot_bars - 2] )
                        ret = BC_RIGHT_TO_LEFT;
        }
        return ret;
}

/*              If  4 consecutive points of the bc data array show descending order   */
```

```
/*      function returns  1, otherwise 0 */

static UCHAR Edge_IsFalling(UCHAR x0,UCHAR x1,UCHAR x2,UCHAR x3)
{
UWORD s = 0;

        if( x1 < x0 )    s += x0 - x1;
        else
                return 0;
        if( x2 < x1 )    s += x1 - x2;
        else
                return 0;
        if( x3 < x2 )    s += x2 - x3;
        else
                return 0;

        if( s > 10)
                return 1;
        else
                return 0;
}

/*              If 4 consecutive points of the bc data array show ascending order   */
/*      function returns  1, otherwise 0 */

static UCHAR Edge_IsRising(UCHAR x0,UCHAR x1,UCHAR x2,UCHAR x3)
{
UWORD s = 0;

        if( x1 > x0 )    s += x1 - x0;
        else
                return 0;
        if( x2 > x1 )    s += x2 - x1;
```

```
        else
                return 0;
        if( x3 > x2 )   s += x3 - x2;
        else
                return 0;

        if( s > 10) return 1;
        else return 0;
}
/*                                                  r| _  |f              */
/*      This function is called at the beginning of the falling edge f.   / \           */
/*      It reverses search until the end of the previous rising edge r.          /  \    */
/*      The  max value found on the way back represents peak's max value   /      \
                */
/*              The dip's width is calculated at the same time
                                        */

static void Get_Peak_Value( UWORD start_ix )
{
UCHAR max_val, p0, p1, p2, p3;
UWORD ix, temp1, temp2;

        max_val = 0;
        for(ix = start_ix; ix > last_r_start_ix  ; ix--)
        {
                p0 = bc_buffer_ptr[ix   ];
                p1 = bc_buffer_ptr[ix - 1];
                p2 = bc_buffer_ptr[ix - 2];
                p3 = bc_buffer_ptr[ix - 3];

                if(p0 > max_val) max_val = p0;
```

```
            if(Edge_IsFalling( p0, p1, p2,p3 ) )
            {
                prev_r_end_ix = last_r_end_ix;
                last_r_end_ix = ix;

                temp1=( last_f_end_ix + prev_f_start_ix )/2 ;
                temp2=(          ix + last_r_start_ix )/2 ;

                Dp_Width[ rise_cnt - 1] = (UCHAR)((temp2 > temp1) ? (temp2 -
temp1) : 1 );
                break;
            }
        }
        Pk_Value[rise_cnt-1] = max_val;
   if( max_val >= Max_High) Max_High = max_val;
}

/*      This function is called at the beginning of the rising edge r.   \        /      */
/*      It reverses search until the end of the previous falling edge f.  \    /         */
/*      The  max value found on the way back represents dip's max value     \  /
        */
/*              The peak's width is calculated at the same time
          f| - |r          */

static void Get_Dip_Value( UWORD start_ix)
{
UCHAR min_val, p0, p1, p2, p3;
UWORD ix, temp1, temp2;

        min_val = 250;
        for(ix = start_ix; ix > last_f_start_ix  ; ix--)
        {
            p0 = bc_buffer_ptr[ix   ];
```

```
            p1 = bc_buffer_ptr[ix - 1];
            p2 = bc_buffer_ptr[ix - 2];
            p3 = bc_buffer_ptr[ix - 3];

            if(p0 < min_val) min_val = p0;

            if( Edge_IsRising( p0, p1, p2, p3 ) )
            {
                  prev_f_end_ix = last_f_end_ix;
                  last_f_end_ix = ix;
                  if(fall_cnt > 1)
                  {
                        temp1=( last_r_end_ix + prev_r_start_ix )/2 ;
                        temp2=(       ix + last_f_start_ix )/2 ;
                        Pk_Width[fall_cnt - 2] = (UCHAR)((temp2 > temp1) ?
(temp2 - temp1) : 1);
                  }
                  break;
            }
      }
      Dp_Value[fall_cnt-1] = min_val;
      if( min_val <= Min_Low ) Min_Low = min_val;
      tot_bars += 1;
}

/*          Function scans the bc data buffer in search of bc pulses.
      */
/*    Calls to Get_Peak_Value and Get_Dip_Value fill all 4 arrays          */
/*          repesenting pulse's width and amplitude
                  */
/*  Dp_Value, Dp_Width, Pk_Value, Pk_Width
                  */
```

```
static void Find_Dips_And_Peaks(void)
{
UWORD      i, up_cnt, dn_cnt,    flat_cnt;
UCHAR   x3, x2, x1, x0;

        if( bc_error_flag > 0 ) return;

        up_cnt              = 0;
        dn_cnt              = 0;
        flat_cnt        = 0;

        for( i = 10; i < bc_data_samples; i++ )
        {
                x0 = bc_buffer_ptr[i  ];
                x1 = bc_buffer_ptr[i+1];
                x2 = bc_buffer_ptr[i+2];
                x3 = bc_buffer_ptr[i+3];

                /* reset not valid counts  */
                if((flat_cnt > MAX_FLAT_POINTS) && (fall_cnt < MIN_FALL_CNT ))
                {
                        up_cnt              = 0;
                        dn_cnt              = 0;
                        Clear_BC_Data();
                }

                /* if bc signal is flat for at least MAX_FLAT_POINTS points, exit search
*/
                if((flat_cnt > MAX_FLAT_POINTS ) && (fall_cnt >= MIN_FALL_CNT ))
break;

                if( Edge_IsFalling( x0, x1, x2, x3 ))
                {
```

```
                flat_cnt        = 0;
                up_cnt                  = 0;
                if(dn_cnt  == 0)                /* start of the falling edge */
                {
                        prev_f_start_ix = last_f_start_ix;
                        last_f_start_ix = i;
                        fall_cnt++;
                        if(rise_cnt > 0)        Get_Peak_Value( i );

                }
                dn_cnt++;
        }
        else
        if( Edge_IsRising( x0, x1, x2, x3 ))
        {
                flat_cnt        = 0;
                dn_cnt                  = 0;
                if(up_cnt  == 0)        /* start of an rising edge     */
                {
                        prev_r_start_ix = last_r_start_ix;
                        last_r_start_ix = i;
                        rise_cnt++;
                        if( fall_cnt > 0 )      Get_Dip_Value( i );

                }
                up_cnt++;
        }
        else
                flat_cnt++;

        if( tot_bars > MAX_FALL_CNT ) /* 2 dips start, 2 dips end 60 bars max
data   */
        {
```

```
                bc_error_flag += 1;
                return;
            }
        }
        dig_bars = (UCHAR)(tot_bars - 4);

        if( (dig_bars % 5) != 0)
            bc_error_flag += 1;

        decodecharcount = (UCHAR)((dig_bars / 5) * 2);

        bc_order = Get_BC_Data_Order();
}
/*  Function searches the five element Dp_WxA and Pk_WxA arrays for 3 greatest
values                          */
/*      The relative position of the two greatest is marked in bar, and spc byte
        */

static void Get_3_Greatest(UWORD* dmax_arr,UWORD* pmax_arr, UCHAR* bar,
UCHAR* spc)
{
UWORD           max_dv,     max_pv,     dip_v, pik_v;
UCHAR           byte_mask, i, run, dix, pix;
UCHAR           bar_b, spc_b  ;

        byte_mask   = (UCHAR)( ( bc_order == BC_LEFT_TO_RIGHT ) ? 0x01 :
0x10 );

        bar_b = 0;
        spc_b = 0;

        for( run = 0; run < 3; run++)
        {
```

```
            max_dv      = 0;    max_pv      = 0;
            dix     = 0;    pix     = 0;
            for( i = 0; i < 5; i++ )
            {
                dip_v = Dp_WxA[i ];
                if(dip_v >= max_dv)
                {
                    max_dv      = dip_v;
                    dix = i;
                }
                pik_v = Pk_WxA[i ];
                if(pik_v >= max_pv)
                {
                    max_pv      = pik_v;
                    pix = i;
                }

            }
            *(dmax_arr++) = Dp_WxA[dix];
            *(pmax_arr++) = Pk_WxA[pix];
            if( run < 2 ) /* only two greatest values are treated as wide pulses   */
            {
                Dp_WxA[dix] = WIDE_PULSE;
                bar_b |= ((bc_order==BC_LEFT_TO_RIGHT) ?
(byte_mask<<(4-dix)) : (byte_mask>>(4-dix)));

                Pk_WxA[pix] = WIDE_PULSE;
                spc_b |= ((bc_order==BC_LEFT_TO_RIGHT) ?
(byte_mask<<(4-pix)) : (byte_mask>>(4-pix)));
            }
        }
        *bar = bar_b;
        *spc = spc_b;
```

```
}

/*  Function traverses all four arrays in 5 dips/pulses steps, loads WxA array      */
/*  with value representing width * amplitude, then searches it for 3  greatest      */
/*  values. The two of them represent wide bar or space, the proximity of the third*/
/*  one is used to determine close cases that should disqualify the result.   */

static void Find_And_Decode2of5(void)
{
UWORD           dv_max[3];    /* dip value max array holds 3 greatest bar values
*/
UWORD           pv_max[3];    /*      peak value max array holds 3 greatest
space values  */
UWORD           d_12, d_23, d_13,   p_12, p_23, p_13;
UCHAR           ix,  pair, pk_st_ix;
UCHAR           bar_byte, spc_byte, dig_ix, err_cnt;

        if( bc_error_flag > 0) return;

        err_cnt         = 0;
        /* pk_st_ix - peak start index depends on BC order, because   */
        /* if LEFT_TO_RIGHT it is preceeded by 2 start peaks (spaces) */
        /* if RIGHT_TO_LEFT it is preceeded by 1 end peaks (space)    */
        pk_st_ix        = (UCHAR)(( bc_order == BC_LEFT_TO_RIGHT ) ? 2 : 1) ;
        dig_ix          = (UCHAR)(( bc_order == BC_LEFT_TO_RIGHT ) ? 0 : ( (
dig_bars/5 )*2 - 1)) ;

        for( pair = 0; pair < dig_bars; pair += 5)
        {
                for(ix = 0; ix < 5; ix++)
                {
                        Dp_WxA[ix] = (UWORD)(Dp_Width[ix+pair+2]  ) *   ( Max_High -
(UWORD)(Dp_Value[ix+pair+2] ));
```

```
                Pk_WxA[ix] = (UWORD)(Pk_Width[ix+pair+pk_st_ix]) *
((UWORD)(Pk_Value[ix+pair+pk_st_ix])-Min_Low);
        }
        Get_3_Greatest(dv_max, pv_max,&bar_byte, &spc_byte);
        /* eliminate close cases when questionable pulses were selected */
        /* 0 first max, 1 second, 2 third
    */
        d_12 = dv_max[0] - dv_max[1];
        d_23 = dv_max[1] - dv_max[2];
        d_13 = dv_max[0] - dv_max[2];

        p_12 = pv_max[0] - pv_max[1];
        p_23 = pv_max[1] - pv_max[2];
        p_13 = pv_max[0] - pv_max[2];

        if( ( (dv_max[0]/4) > d_13 ) || ( d_12/4 > d_23  ) )
        {
            err_cnt += 1;
            if(err_cnt >= 2)
            {
                bc_error_flag += 1;
                return;
            }
        }

        if( ( (pv_max[0]/4 ) > p_13 ) || ( p_12/4 > p_23  ) )
        {
            err_cnt += 1;
            if(err_cnt >= 2)
            {
                bc_error_flag += 1;
                return;
            }
```

```
            }
            if(     bc_order == BC_LEFT_TO_RIGHT )
            {
                    output_buffer[ dig_ix++ ] = Decode_Digit( bar_byte ) ;
                    output_buffer[ dig_ix++ ] = Decode_Digit( spc_byte ) ;
            }
            else
            {
                    output_buffer[ dig_ix-- ] = Decode_Digit( spc_byte ) ;
                    output_buffer[ dig_ix-- ] = Decode_Digit( bar_byte ) ;
             }
        }
}
/*      The Interleave_2of5 bar code uses wide (w) and narrow (n) bars and spaces
            */
/*          to represent digits.
                                                    */
/*      In this module "nnwwn" that stands for '0' is shown as a binary 00110 (hex
0x06). */
/*      The "wnnnw" combination that stands for '1' is binary 10001 (hex 0x11) etc.
            */

static UCHAR Decode_Digit(UCHAR dig)
{
UCHAR ch;
        switch(dig)
        {
                case 0x06 :  ch = '0';      break;
                case 0x11 :  ch = '1';      break;
                case 0x09 :  ch = '2';      break;
                case 0x18 :  ch = '3';      break;
                case 0x05 :  ch = '4';      break;
                case 0x14 :  ch = '5';      break;
```

```
            case 0x0C :  ch = '6';       break;
            case 0x03 :  ch = '7';       break;
            case 0x12 :  ch = '8';       break;
            case 0x0A :  ch = '9';       break;
            default   :  ch = 0x00;      bc_error_flag += 1;
                                         break;
        }
        return ch;
}
/*
  This function controls the Barcode decoding process.
*/
UCHAR gBarcode_Process(recog_result_t *rptr)
{
   UCHAR        removal;
   recog_result_t  *xptr;

   xptr = rptr;
   /* make sure we are decoding for BARCODE      */
   if (rptr->currenttype != (UCHAR)GBT_COUPON_TYPE) return (end_pass);
        /* if more than one candidate, assume the other candidate is a note, */
        /* so this cannot be a barcode  */
   if (rptr->candidates != 1)
   {
      removal = Recog_RemoveCandidate();   /* delete the current candidate */
      return (end_pass);
   }

        /* first take care of upper bc reader data   */

        Clear_BC_Data();

        bc_error_flag        = 0;
```

```
  bc_data_samples = bc_num_of_pts;                    /* length of the first bcr
buffer  */

  bc_buffer_ptr     = gDatacoll_BarCodeBuffer;        /* pointer to the first bcr
buffer */

      memset(output_buffer, 0, MAX_BC_DIGITS);

      Find_Dips_And_Peaks();

      Find_And_Decode2of5();

  /*  if no error_flag,  its good; otherwise remove the barcode  as a candidate  */
  if(  bc_error_flag > 0 )
  {
      Clear_BC_Data();

      bc_error_flag        = 0;

      bc_data_samples = bc_num_of_pts2;               /* length of the second
bcr buffer  */

      bc_buffer_ptr = gDatacoll_BarCodeBuffer2 ;  /* pointer to the second bcr
buffer */

      memset(output_buffer, 0, MAX_BC_DIGITS);

      Find_Dips_And_Peaks();

            Find_And_Decode2of5();

      if(  bc_error_flag > 0 )
```

```
        {
                removal = Recog_RemoveCandidate();  /* delete the current candidate
*/
        }

    }
    return(end_pass);
}

/* public function used to initialize variables and pointers to decode the barcode  */
UCHAR gBarcode_Init ( void **nptr)
{
        memset(output_buffer, 0, MAX_BC_DIGITS);

        bc_error_flag  = 0;

        Clear_BC_Data();

    nltptr = (void*)nptr;      /* assign the local pointer            */

    return (continue_pass);
}
```

[037] First signal source 15 and/or second signal source 20 may be any known signal source known in the art, for example, a light source, a light-emitting diode (LED), an infrared LED, a red LED, and/or a blue LED. First signal detector 16 and/or second signal detector 19 may be any known signal detector, for example, a light detector, an LED light detector, an infrared light detector, and/or a blue light detector.

[038] Information source 200 may be any known device and/or material capable of carrying information, for example, paper(s), plastic(s), bank note(s), a computer chip(s), and/or metal(s. Information source 200 may have any desired

size, shape, and/or configuration, for example, a substantially rectangular paper receipt.

[039] Information 201 may be any information, for example, barcode(s), text, picture(s), personal information, and/or computer code(s). Information 201 may be disposed on information source 200 using any known method, for example, ink(s), chemical(s), watermark(s), depression(s), and/or protrusion(s). Information 201 may be disposed on any portion of information source 200. For example, as shown in Figs. 3A and 3B, information 201a may be disposed on one side of information source 200, while information source 201b may be disposed on another side of information source 200. Information 201a, 201b may be disposed on substantially opposing portions of information source 200 and/or may be disposed on longitudinally offset portions. Information 201a may be the same as information 201b, or may be different. In another example, information 201 may be disposed on only side of information source 200.

[040] Information source 200 may be transported through validator 100 such that at least one of information readers 10, 11 may read information 201 disposed on information source 200. Validator 100 may be configured such that at least one of information readers 10, 11 may read information 201 disposed on information source 200 as information source 200 is being transported through validator 100. For example, information readers 10, 11 may be disposed on opposite sides of insertion channel 2. In another example, information readers 10, 11 may be disposed such that each of information readers 10, 11 may simultaneously read information 201 from information source 200. In a further example, information readers 10, 11 may be disposed such that information readers 10, 11 cannot simultaneously read information 201 from information source 200. In yet another example, information readers 10, 11 may be staggered and/or offset in a longitudinal direction of insertion channel 2 and/or a transport direction of information source 200, for example, as shown in Fig. 1. This may assist in preventing signals from information readers 10, 11 from interfering with each other. In a yet further example, information readers 10, 11 may be substantially disposed across from each other around insertion channel 2. In still another example, information readers 10, 11, may be disposed substantially parallel to each other. In a still further example, information readers 10, 11, may be disposed at an angle relative to each other.

Other configurations of information readers 10, 11 relative to each other are contemplated. Information readers 10, 11, may be configured to continuously operate and/or operate only when information source 200 is positioned such that information readers 10, 11 can read information 201 from information source 200.

[041] Once information source 200 has cleared information readers 10, 11, information source 200 may cease to block the signal from first signal source 15 to first signal detector 16. Accordingly, first signal detector 16 may inform microprocessor 17 that it is no longer receiving a signal or is receiving a signal if previously blocked from first signal source 15 (e.g., by sending a signal or ceasing to send a signal), and in turn microprocessor 17 may deactivate actuator 3. The deactivation of actuator 3 may then cause one or more of gear(s) 4, 5, 6, pulley(s) 7, drive belts 8, 9, roller(s) 12, and drive wheel(s) 14 to cease thereby ceasing the transport of information source 200. At this point, information source 200 may be held in validator 100 in an escrow position.

[042] Before, during, or after information source 200 may be held in an escrow position, microprocessor 17 may be processing information 201 and/or other information sent by information readers 10, 11. For example, microprocessor 17 may determine whether one or more of information 201 and/or other information is valid information 201. If so, information 201 may be sent to another microprocessor 317 disposed in apparatus 300, for example, a gambling machine. In another example, microprocessor 17 may determine whether one or more of information 201 and/or other information is indeed the same information or related information. Microprocessor 17, 317 may also or alternatively process the information 201 and/or other information using any known algorithm.

[043] Microprocessor 17, 317 may transmit the information to central processor 417, for example, a casino's central processor. Central processor 417 may process information 201 using any known algorithm, for example, verify information 201 and then transmit to microprocessor 17, 317 one or more stored values related to information 201, for example, monetary value information. Microprocessor 17, 317 may then signal validator 100 to accept and/or store information source 200. Microprocessor 17, 317 may then display the one or more stored values on apparatus 300, for example, providing credits on the slot machine so that the user may play.

[044] Central processor 417 may also or alternatively decide that information 201 is incorrect, fraudulent, and/or not verified, in which case central processor 417 may send a signal to microprocessor 17, 317 to reject the information source, in which case microprocessor 17, 317 may cause validator 100 to either eject information source 200 and/or store information source 200 without allowing a user to operate apparatus 300.

[045] In another embodiment, validator 100 may be used to read information 201a, 201b that is disposed on two or more portions of information source 200. For example, information 201 may be a barcode number 201 that is too long to be placed on one side of an information source 200 (e.g., barcode number 201 may have more than about 20 characters). In that case, information 201 may be broken up into at least two information portions 201a, 201b and be placed on two or more portions of information source 200, for example, opposite sides of information source 200. Validator 100 may then be used to determine information 201, for example, information readers 10, 11 may each read one of information portions 201a, 201b. Information readers 10, 11 may each read either of information portions 201a, 201b. At least one of information portions 201a, 201b may include additional information different from information 201 that may assist one of microprocessor 17, 317, 417 in determining how information portions 201a, 201b are related, for example, which of barcode number portions 201a, 201b goes "first" to create longer barcode number 201. Such longer barcode numbers 201 may be desirable, for example, to prevent the recycling of barcode numbers.

[046] Embodiments of the invention may include a method of reading information 201 from an information source 200 using validator 100, for example, as shown in Fig. 5. In an example where information 201 is disposed on only one side of information source 200, a user may place information source 200 including information 201 in insertion channel 2. Information source 200 may interrupt a signal between first signal source 15 and first signal detector 16, causing actuator 3 to activate, for example, via first signal detector 16 either sending a signal or ceasing to send a signal to microprocessor 17. Actuator 3 may actuate one or more of gear(s) 4, 5, 6, pulley(s) 7, drive belts 8, 9, roller(s) 12, and drive wheel(s) 14 and cause information source 200 to be transported further into validator 100.

[047] While information source 200 is being transported through validator 100, information source 200 may be sent past information readers 10, 11. One of information readers 10, 11 may read information 201 from one side of information source 200. To read information 201, second signal source 20 from one of information readers 10, 11 may send a signal that may be reflected, altered, and/or changed by information 201 and then detected by second signal detector 19. Second signal detector 19 may then send a signal to microprocessor 17. In the alternative, signal detector 19 may send continuous signals to microprocessor 17, in which case microprocessor 17 may determine which of the signals constitutes information 201. When microprocessor 17 determines that is has received information 201, microprocessor 17 may process information 201 and/or send information 201 to microprocessor 317 in apparatus 300.

[048] Microprocessor 17, 317 may process information 201 using any known algorithm, for example, if information 201 is a barcode, determine the barcode number. Microprocessor 17, 317 may then send information 201 to central processor 417. Central processor 417 may then associate information 201with stored values, for example, a monetary value information and/or personal information. Central processor 417 may then send one or more of those stored values back to microprocessor 17, 317, and then microprocessor 17, 317 may display the stored values on apparatus 300, for example, display monetary value information on a slot machine. Microprocessor 17, 317 may also at that point allow use of apparatus 300 and/or store information source 300. Central processor 417 may in the alternative determine that information 201 is not valid, and thus send a signal to microprocessor 17, 317 to either reject or retain information source 200. In such a case, apparatus 300 could not be used.

[049] In an example where information 201a, 201b is disposed on both sides of information source 200, a user may place information source 200 including information 201a, 201b in insertion channel 2. Information source 200 may interrupt a signal between first signal source 15 and first signal detector 16, causing actuator 3 to activate, for example, via first signal detector 16 either sending a signal or ceasing to send a signal to microprocessor 17. Actuator 3 may actuate one or more of gear(s) 4, 5, 6, pulley(s) 7, drive belts 8, 9, roller(s) 12, and drive wheel(s) 14 and cause information source 200 to be transported further into validator 100.

[050] While information source 200 is being transported through validator 100, information source 200 may be sent past information readers 10, 11. Each of information readers 10, 11 may read one of information 201a, 201b. Depending on the configuration of information readers 10, 11, information readers 10, 11 may read their respective information 201a, 201b at the same time or at different times. For example, if information 201a, 201b are disposed at substantially corresponding points on opposite sides of information source 200, but information readers 10, 11 are offset in a direction of travel of information source 200 through validator 100, than information readers 10, 11 may read their respective information 201a, 201b at different times. Other configurations or information readers 10, 11 in validator 100 and/or information 201a, 201b on information source 200 are contemplated.

[051] To read information 201, second signal source 20 from one of information readers 10, 11 may send a signal that may be reflected, altered, and/or changed by one of information 201s, 201b and then detected by second signal detector 19. Second signal detector 19 may then send a signal to microprocessor 17 corresponding to one of information 201a, 201b. In the alternative, signal detector 19 may send continuous signals to microprocessor 17, in which case microprocessor 17 may determine which of the signals constitutes information 201a, 201b. When microprocessor 17 determines that is has received information 201a, 201b, microprocessor 17 may process information 201a, 201b and/or send information 201a, 201b to microprocessor 317 in apparatus 300.

[052] Microprocessor 17, 317 may process information 201a, 201b using any known algorithm. For example, if information 201a, 201b (e.g., barcode portions) are different portions of a single piece of information 201 (e.g., a barcode), microprocessor 17, 317 may combine information 201a, 201b into single piece of information 201. One of information 201a, 201b may include additional information that indicates to microprocessor 17, 317 how to combine information 201a, 201b into single piece of information 201. Microprocessor 17, 317 may then send information 201 to central processor 417. Central processor 417 may then associate information 201 with stored values, for example, a monetary value information and/or personal information. Central processor 417 may then send one or more of those stored values back to microprocessor 17, 317, and then microprocessor 17, 317 may display the stored values on apparatus 300, for example, display monetary value

information on a slot machine. Microprocessor 17, 317 may also at that point allow use of apparatus 300 and/or store information source 300. Central processor 417 may in the alternative determine that information 201 is not valid, and thus send a signal to microprocessor 17, 317 to either reject or retain information source 200. In such a case, apparatus 300 could not be used.

[053] Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

1. A validator, comprising:

at least two information readers configured to read information from an information source; and

a transporter configured to physically transport the information source past the at least two information readers such that at least one of the at least two information readers reads information from the information source.

2. The validator of claim 1, wherein the at least two information readers are configured to read information from different sides of the information source.

3. The validator of claim 1, wherein the at least two information readers are offset from each other in a direction of transport of the information source.

4. The validator of claim 1, further comprising a housing,

wherein the at least two information readers are disposed in the housing such that each of the at least two information readers can only read information disposed on one side of the information source.

5. The validator of claim 1, further comprising a housing,

wherein the at least two information readers are disposed in the housing such that a first of the at least two information readers can only read information disposed on a first side of the information source and a first of the at least two information readers can only read information disposed on a second side of the information source different from the first side.

6. A system including the validator of claim 1.

7. A validator, comprising:

a first information reader configured to read a first information portion disposed on a first portion of an information source;

a second information reader configured to read a second information portion disposed on a second portion of the information source different from the first portion;

a transporter configured to physically transport the information source past the first information reader and the second information reader such that the first information reader reads the first information portion from the first portion of the information source and the second information reader reads the second information portion from the second portion of the information source; and

a microprocessor configured to combine the first information portion and a second information portion into a third information portion.

8. The validator of claim 7, further comprising a housing,

wherein the first information portion is disposed in the housing on a first side of the information source and the second information portion is disposed in the housing on a second side of the information source different from the first side.

9. The validator of claim 7, further comprising a housing,

wherein the at least two information readers are offset from each other in the housing in a direction of transport of the information source.

10. The validator of claim 7, wherein the first information portion is a first barcode portion, the second information portion is a second barcode portion, and the third information portion is a complete barcode.

11. The validator of claim 10, wherein the complete barcode has more than 20 characters.

12. A system including the validator of claim 7.

13. A method, comprising:

providing a validator including at least two information readers;

inserting an information source into the validator;

transporting the information source past the at least two information readers; and

reading information from the information source via at least one of the at least two information readers.

14. The method of claim 13, further comprising reading information from the information source via both of the at least two information readers.

15. The method of claim 13, further comprising reading a first information portion via a first of the at least two information readers and reading a second information portion via a second of the at least two information readers.

16. The method of claim 13, further comprising reading a first information portion from a first side of the information source via a first of the at least two information readers and reading a second information portion from a second side of the information source via a second of the at least two information readers.

17. The method of claim 13, wherein a first information portion is read by a first of the at least two information readers during a first time interval and a second information portion is read by a second of the at least information readers during a second time interval different from the first time interval.

18. A method, comprising:

providing a validator including a first information reader and a second information reader;

inserting an information source into the validator;

transporting the information source past the first information reader and the second information reader;

reading a first information portion from the information source via the first information reader and a second information portion from the information source via the second information reader; and

combining the first information and the second information portion into a third information portion.

19. The method of claim 18, wherein the first information portion is disposed on a first side of the information source and the second information portion is disposed on a second side of the information source different from the first side.

20. The method of claim 18, wherein the first information portion is read by the first information reader during a first time interval and the second information portion is read by the second information reader during a second time interval different from the first time interval.

21. The method of claim 18, wherein the first information portion is a first barcode portion, the second information portion is a second barcode portion, and the third information portion is a complete barcode.

22. The method of claim 18, wherein the complete barcode has more than 20 characters.

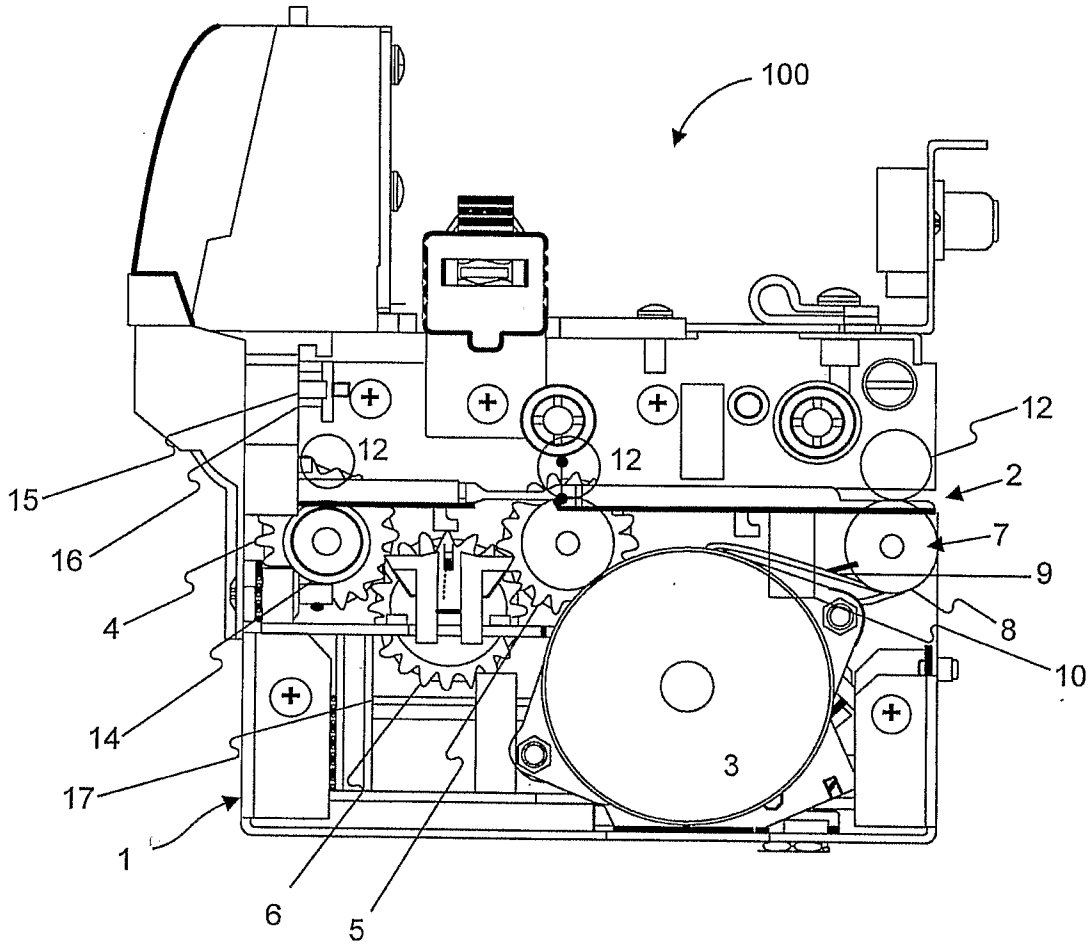

FIG. 1

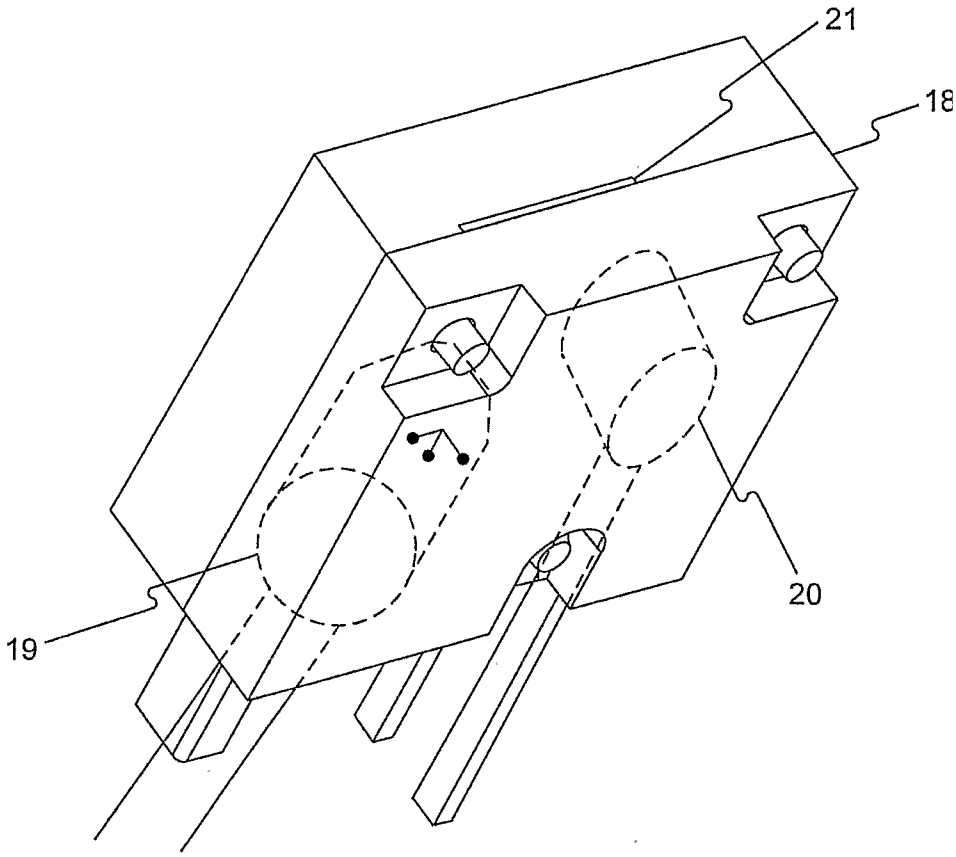

FIG. 2

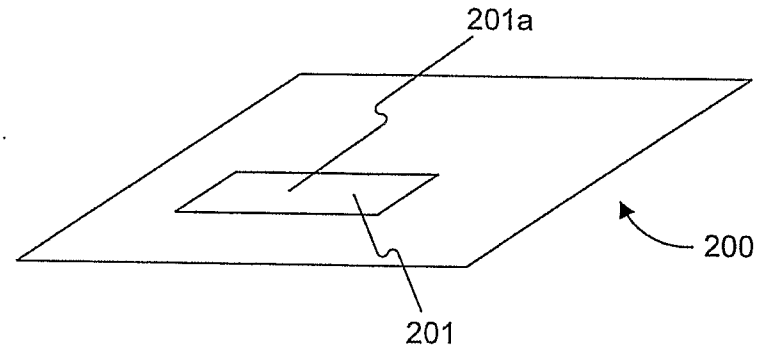

FIG. 3A

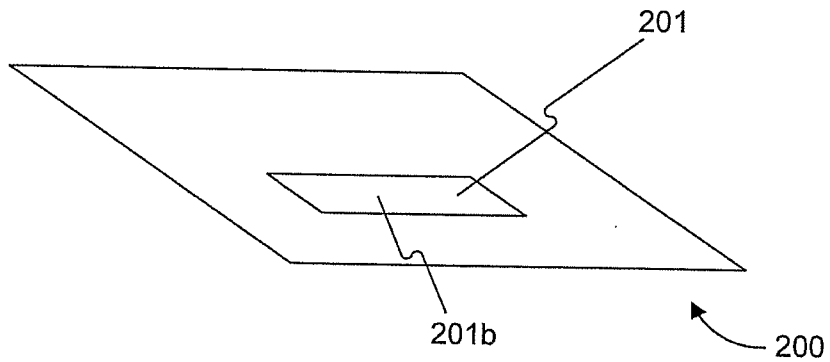

FIG. 3B

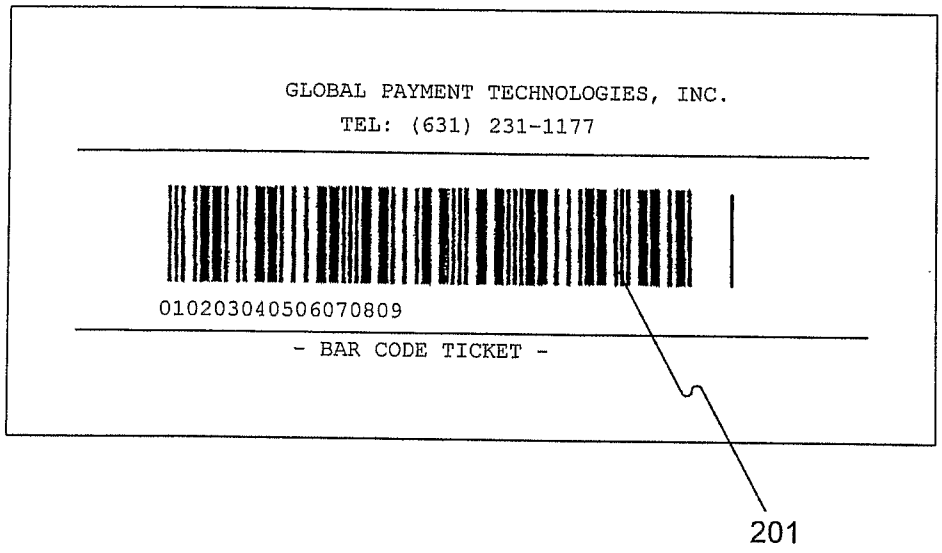
GLOBAL PAYMENT TECHNOLOGIES, INC.
TEL: (631) 231-1177
010203040506070809
- BAR CODE TICKET -
200
201

FIG. 3C

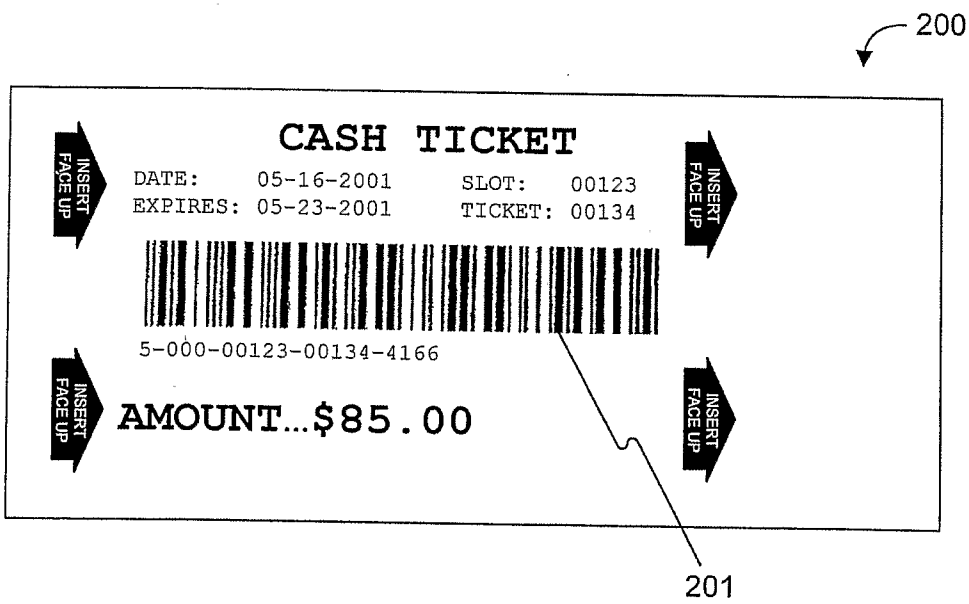
200
CASH TICKET
DATE: 05-16-2001
EXPIRES: 05-23-2001
SLOT: 00123
TICKET: 00134
INSERT FACE UP
INSERT FACE UP
INSERT FACE UP
INSERT FACE UP
5-000-00123-00134-4166
AMOUNT...$85.00
201

FIG. 3D

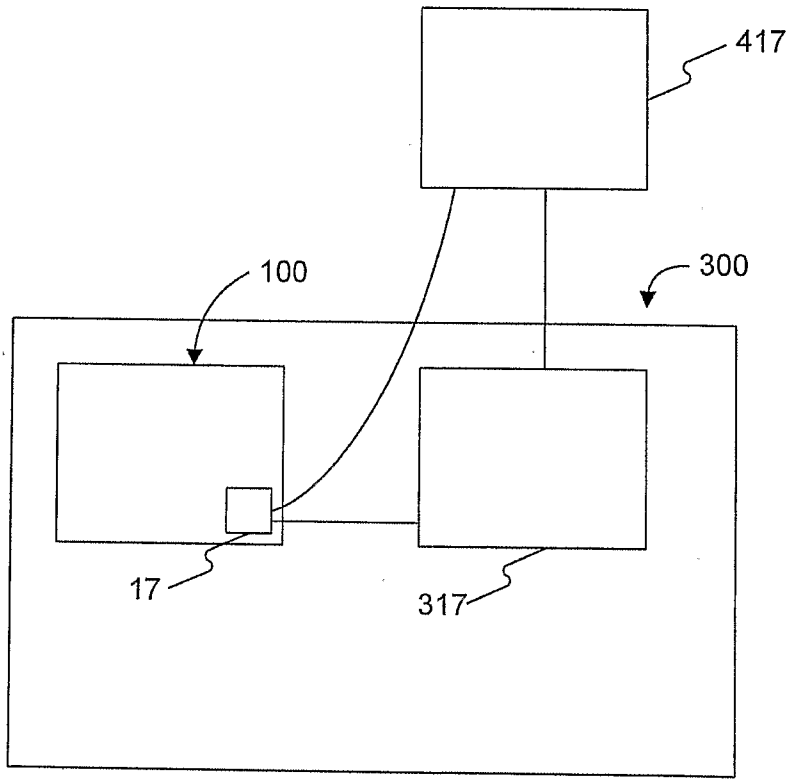

FIG. 4

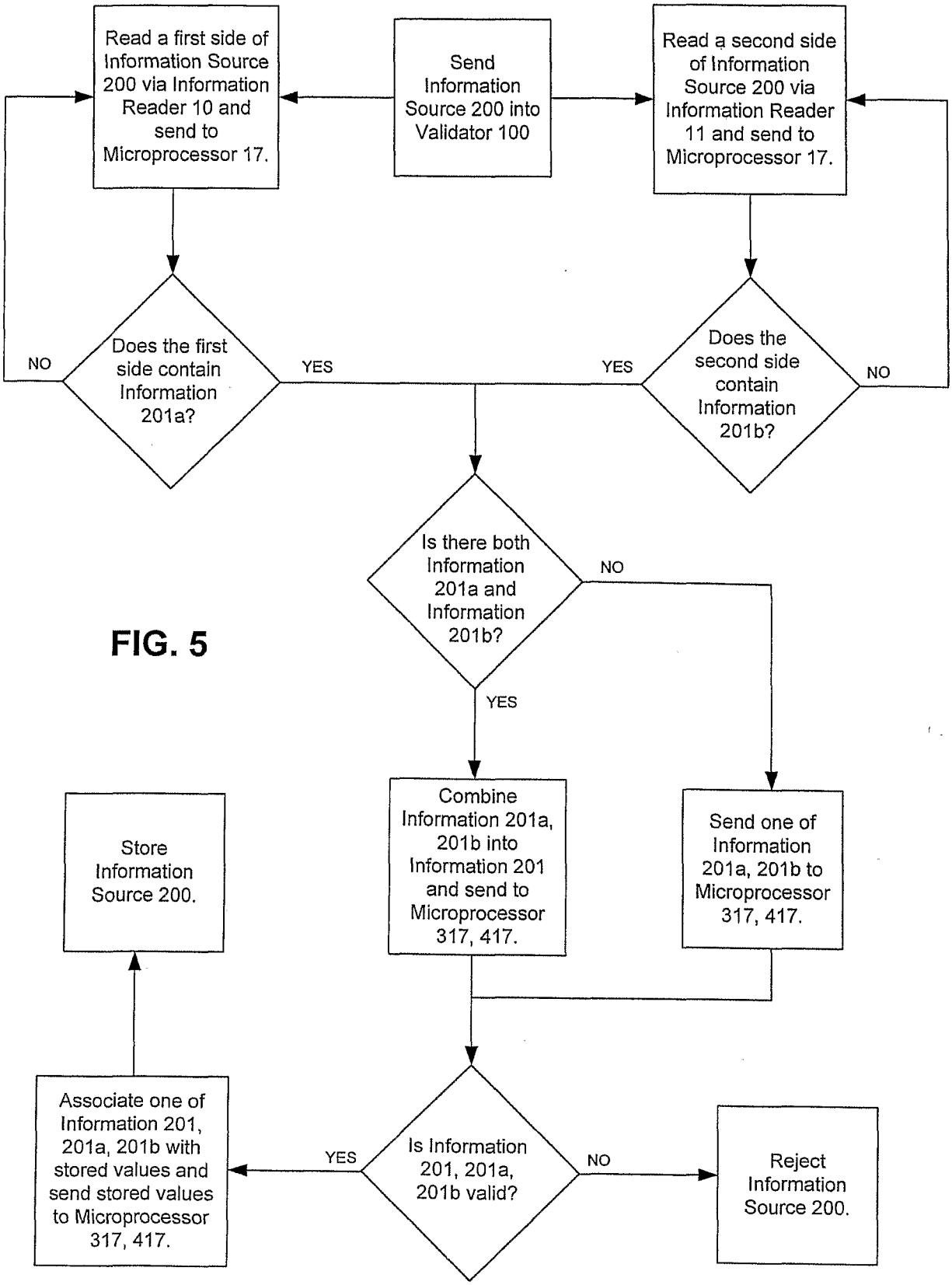
Send Information Source 200 into Validator 100
Read a first side of Information Source 200 via Information Reader 10 and send to Microprocessor 17.
Read a second side of Information Source 200 via Information Reader 11 and send to Microprocessor 17.
Does the first side contain Information 201a?
NO
YES
Does the second side contain Information 201b?
YES
NO
Is there both Information 201a and Information 201b?
NO
YES
Combine Information 201a, 201b into Information 201 and send to Microprocessor 317, 417.
Send one of Information 201a, 201b to Microprocessor 317, 417.
Store Information Source 200.
Associate one of Information 201, 201a, 201b with stored values and send stored values to Microprocessor 317, 417.
YES
Is Information 201, 201a, 201b valid?
NO
Reject Information Source 200.

FIG. 5

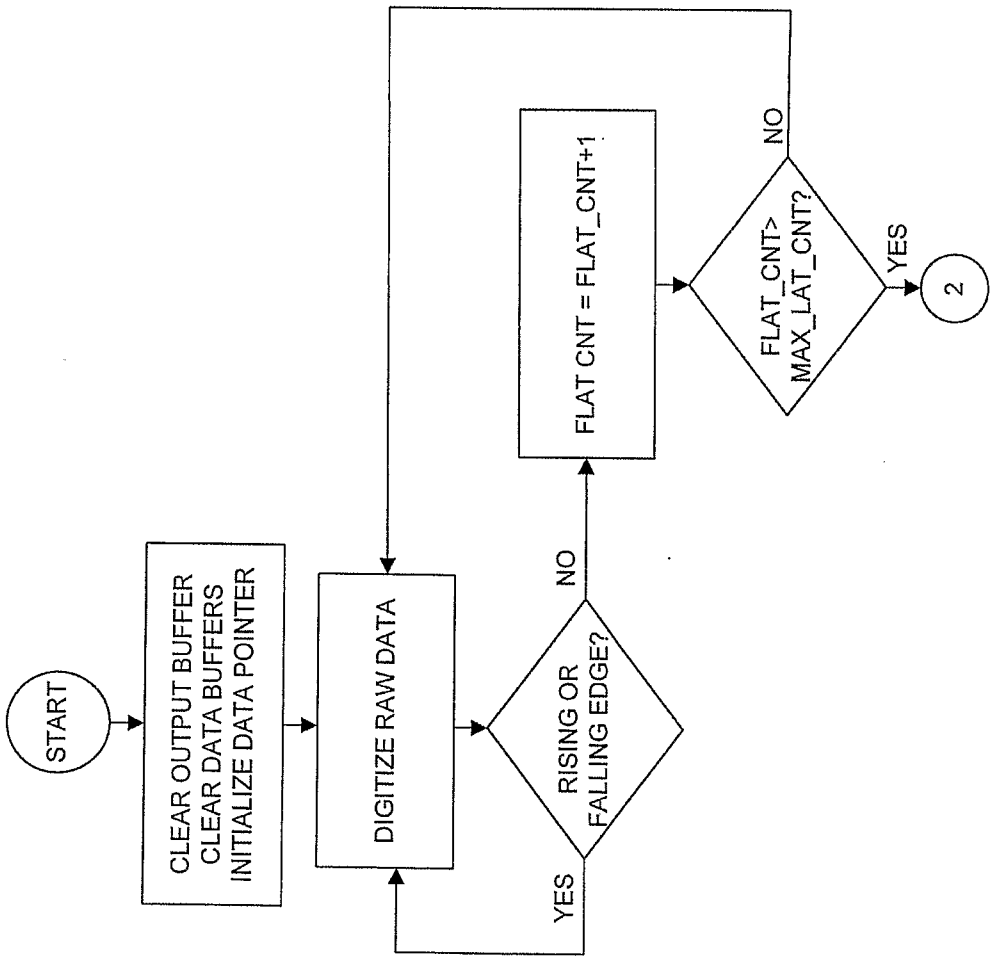
START
CLEAR OUTPUT BUFFER
CLEAR DATA BUFFERS
INITIALIZE DATA POINTER
DIGITIZE RAW DATA
RISING OR FALLING EDGE?
NO
YES
FLAT CNT = FLAT_CNT+1
FLAT_CNT> MAX_LAT_CNT?
NO
YES
2

FIG. 6A

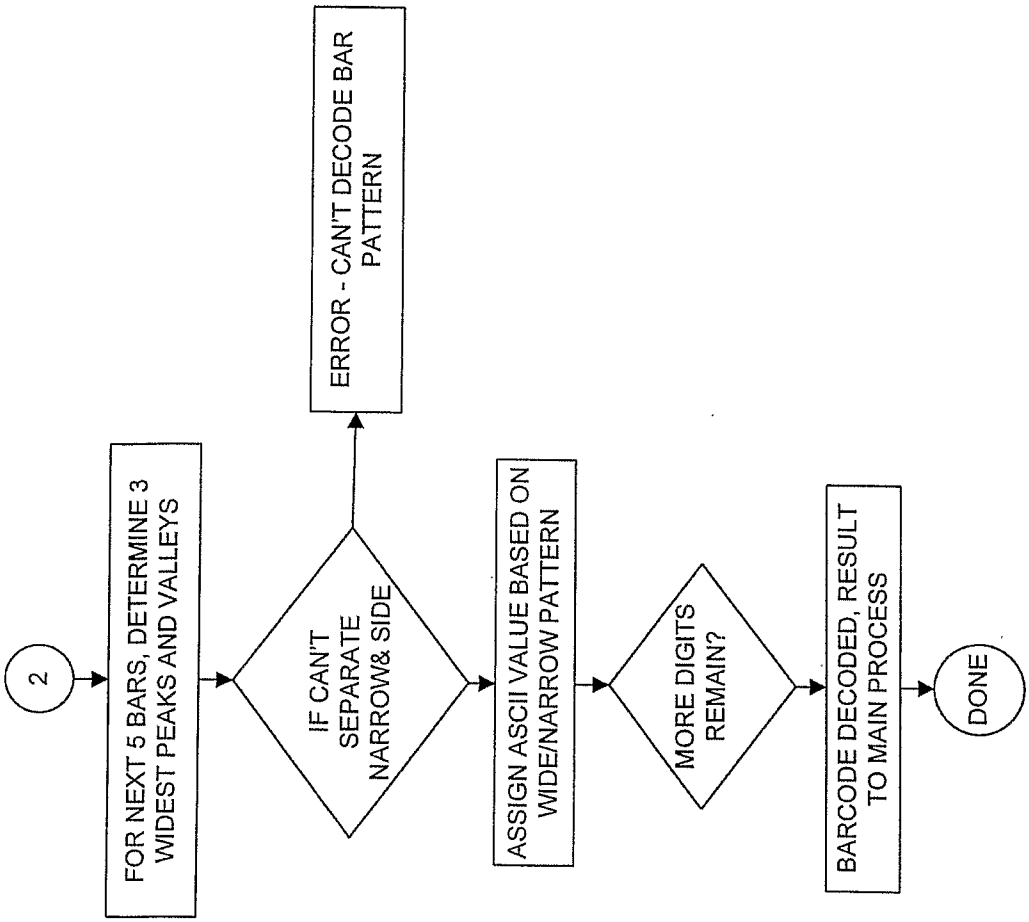
2
FOR NEXT 5 BARS, DETERMINE 3 WIDEST PEAKS AND VALLEYS
IF CAN'T SEPARATE NARROW& SIDE
ERROR - CAN'T DECODE BAR PATTERN
ASSIGN ASCII VALUE BASED ON WIDE/NARROW PATTERN
MORE DIGITS REMAIN?
BARCODE DECODED, RESULT TO MAIN PROCESS
DONE

FIG. 6B