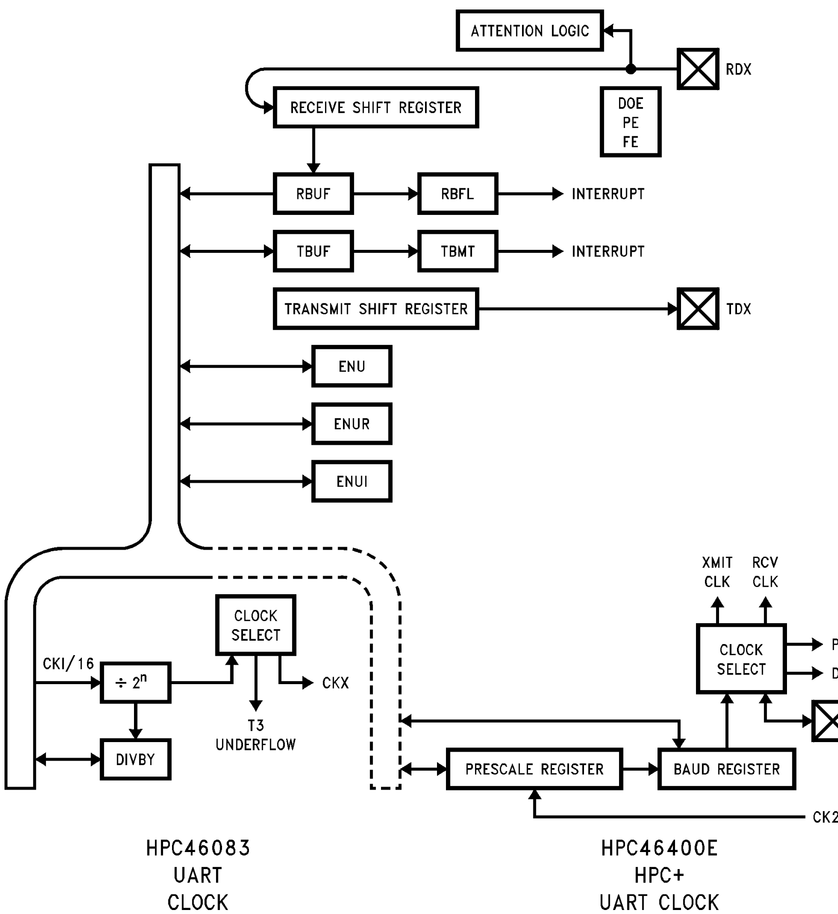
Improved UART Clocking Techniques on New Generation HPCs

National Semiconductor
Application Note 798
Ravi Kumar
April 1993

The new generation HPCs have on-chip UARTs with much better baud rate generation techniques and better status reporting capabilities. This article explains in detail, accurate baud rate generation on HPC46400E and HPC+ UARTs with appropriate examples.

UART implemented on the HPC46400E and HPC+ is an upward compatible enhancement of the UART present on the HPC46083. Unlike the UART on HPC46083, the operating mode may be selected as either Asynchronous or Synchronous. Here we can also select the baud rate through software in conjunction with both prescalar and baud select registers.



TL/DD/11292–1

FIGURE 1

COMMON FEATURES SUPPORTED BY HPC46083 UART AND THE NEWER VERSION OF UART ON HPC46400E AND HPC+

- Fully programmable serial interface characteristics, including:
  - — 8- or 9-bit characters
  - — 1 or 2 stop bits
- Two interrupt sources (Receiver buffer full, Transmit buffer empty)
- Independent clock inputs (either on-chip or off-chip) for the transmitter and receiver
- Error reporting capabilities (Data overrun error, framing error)
- Attention or wake up mode for receiver to enhance networking capability

ADDITIONAL UART FEATURES AVAILABLE ON HPC46400E AND HPC+

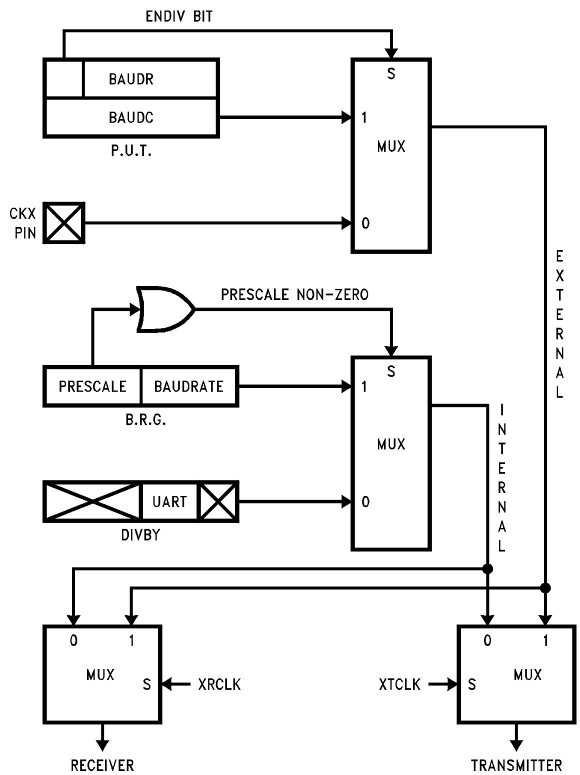
- Upwardly compatible from earlier HPC UARTs such as HPC16083
- Fully programmable serial interface characteristics, including:
  - — Accurate baud rate generation without the penalty of using an expensive crystal up to 625k baud
  - — 7-bit characters possible
  - — 7⁄8, 1 7⁄8 stop bit lengths
  - — Odd, Even, Mark, Space or no parity bit generation and detection
- Selectable Asynchronous or Synchronous mode of operation
- Loopback Diagnostic test capability

Now lets see various methods of BAUD Rate generation.

First we shall discuss how DIVBY can be used to generate required baud-rate.

1.0 UART CLOCK SOURCE FROM DIVBY REGISTER

Clock for DIVBY register can be generated using precise value crystals or T3 underflow. Referring to *Figure 2*, we see that baud rate is from internal source for DIVBY register.



TL/DD/11292–2

FIGURE 2. Simplified UART Clock Routing

The following is a sample assembly language routine illustrating BAUD Rate generation using DIVBY register through precise value crystal.

```
;This program will test the HPC16400E UART for 9600 baud.
;using 10.0 MHz crystal and DIVBY (baud clock from internal source).
;************************************************************
;The power-up default setup is:
;a) Baud clock from internal source DIVBY
;b) Frame format is 1 start, 8 dta, and 1 stop bit.
;The clock should be a 10.0 MHz crystal

.sect uart, rom16

begin:
                sbit  0,0f2,b              ;DIRB reg pin 1 outward direction
                sibit 0,0f4.b              ;BFUNL reg, turns on TDX bit
                rbit  2,0x122.b            ;xtclk
                rbit  3,0x122.b            ;xrclk
                ld    018e.b,#040          ;Load DIVBY from table to generate
                                           ;9600 baud (CKI/64)

;The baud clock = baud rate * 16
;So, for 9600 baud, bclk = 9600 * 16 = 153600 Hz
;With 10.0 MHz clock  →  10.0 MHz/64 = 156250 Hz (within 5%)

xmit:           ld    a,#041               ;load char "A"
                st    a,0126.b             ;Load TBUF reg to transmit
chk:            ifbit 0,0120.b
                jp    xmit                 ;continue to xmit
                jp    chk

.endsect
.end begin
```

Hence we see the percentage error of Baud Rate produced is:

$$\% \text{ error} = (156250 - 153600)/15360$$
$$= 1.72$$

which is within the error limits.

A) Baud Rate Calculation Using DIVBY Register through Precise Value Crystal

Table I gives the bit values to be loaded into UART section of the DIVBY register. This table defines the baud rates for two different crystals at 9.304 MHz and 19.6608 MHz.

We see that more care in selecting the crystal frequency is necessary to generate exact baud rates. Obviously the baud rate generation is restricted by the crystal frequency.

TABLE I

Bit 7	Bit 6	Bit 5	Bit 4	Baud Clock (x16 Clock)	Baud Rate 9.8304 MHz Crystal	Baud Rate 17.6603 MHz Crystal
0	0	0	0	← Not Allowed →		
0	0	0	0	← Defined by Timer T3 Underflow →		
0	0	1	0	CKI/16	38400	65536
0	0	1	1	CKI/32	19200	32768
0	1	0	0	CKI/64	9600	16384
0	1	0	1	CKI/128	4800	8192
0	1	1	0	CKI/256	2400	4096
0	1	1	1	CKI/512	1200	2048
1	0	0	0	CKI/1024	600	1024
1	0	0	1	CKI/2048	300	512
1	0	1	0	CKI/4096	150	256
1	0	1	1	CKI/8192	75	128
1	1	0	0	CKI/16384	38	64
1	1	0	1	CKI/32768	19	32
1	1	1	0	CKI/65536	9.4	16
1	1	1	1	CKI/131072	4.7	8

B) Baud Rate Calculation Using DIVBY Register and Timer T3 Underflow

Suppose we want to generate 9600 baud. In the DIVBY register, load the UART bits with value 0001, which means Baud Clock is defined by T3 underflow (refer to Table I). Once again referring to *Figure 2*, we see BAUD clock is from internal source.

Let's calculate the Pre-Scale value to be loaded into T3 register (0X018C) and R3 register (0X018A)

Baud Clock = Required baud rate × 16

$$\text{Clock Input} = \frac{\text{Crystal Freq}}{16}$$

$$\text{Pre-Scale Value} = \frac{\text{Clock Input}}{\text{Baud Clock}}$$

In our specific case

required BAUD Rate = 9600

crystal freq = 20 MHz

→ Baud Clock = 9600 × 16 = 15360

Clock Input = 20/16 = 1.25 MHz = 1.25×10^6 Hz

$$\text{Pre-Scale Value} = \frac{1.25 \times 10^6}{153600} \approx 8$$

Pre-Scale Value = 8

Actual value to be loaded into T3 and R3 register is (Pre-scale value − 1) i.e., 7 in this case.

Percentage error of Baud Rate produced is:

Pre-Scale Value = 8
Baud Rate = Baud Clock/16
Baud Clock = Clock Input/Pre-Scale Value
Clock Input = CKI/16 = 20 MHz/16
= 1.25 MHz
Baud Clock = 1.25 MHz/8 = 156250
Therefore Baud Rate = 156250/16 = 9765.62

Hence % error = (9765.62 − 9600)/9600
= 1.72

Which is within the error limits.

The following is a sample assembly language routine illustrating BAUD Rate generation through DIVBY and T3 underflow.

```
;******** Generation of BAUD clock through timer3 without triggering timer intrpt ********

.sect uart, rom16

tmr:            ld_tmmode.w,#0xcccc       ;stop timers t3, t2, t1
                ld_divby.w,#0x2010        ;Clk to T3 thru DIVBY as CKI/16
                                          ;with reload val Ton & Toff as 7
                ld_t3reg.w.#0x7           ;Reload Ton as 7
                ld_r3reg.w,#0x7           ;Reload Toff as 7
                                          ;and BAUD rate = 9600
                rbit 2,0x122.b            ;uart internal xmit clk
                rbit 3,0x122.b            ;uart internal rcv clk
                sbit 0,0x0f2.b            ;config BFUN pin as TDX
                sbit 0,0x0f4.b            ;config DIRB pin 1 outward
                ld_tmmode.w,#0x8ccc       ;Start timer T3 & stop T1, T2 & Ack'em.

;Loop to continuously xmit char "A" at specified baud rate

xmit:           ld a,#041                 ;load char "A"
                st a,0126.b               ;load TBUF reg to transmit
chk:            ifbit 0,_enu.b
                jp xmit
                jp chk
.endsect
.end tmr
```

2.0 BAUD RATE CALCULATION USING PUT (PRECISION UART TIMER)

The Precision UART Timer (PUT) is now obsolete and kept only for compatibility with software developed for those earlier components. PUT has two registers i.e., BAUDR with 15-bit divisor field and BAUDC, a 15-bit free-running down counter. These can be programmed to divide the CK2 (CKI/2) clock by a factor of from 3 to 32767, in units of CK2, thus yielding a time base to the UART of higher resolution than that available through the DIVBY register.

Referring to *Figure 2* we see that BAUD clock source for PUT is external.

Suppose the Clock input is 16 MHz and the required baud rate is 9600, then the value to be loaded into BAUDR register will be

$$\text{Required Baud Rate} = \frac{(\text{CK2}/16)}{(\text{BAUDR}+1)}$$

Where CK2 = CKI/2
Given CKI = 16 MHz
Hence CK2 = 8 MHz

$$(\text{BAUDR} + 1) = \frac{\text{CK2}/16}{\text{Required Baud Rate}}$$

$$(\text{BAUDR} + 1) = \frac{8\ \text{MHz}/16}{(9600)}$$

$$\therefore \text{BAUDR} \approx 52 - 1 \qquad 51 \text{ in decimal}$$

and here value to be loaded into BAUDR register will be 33 hex.

Now to select PUT timer as external clock source MSB of BAUDR register must be 1.

1 0 0 0	0 0 0 0	0 0 1 1	0 0 1 1	— Binary
8	0	3	3	— Hex

Note: BAUDC must also be loaded with same value (Reload Value).

Percentage error of Baud Rate produced is:

$$\text{BAUDR} = 51$$

$$\text{Therefore Baud Rate} = \frac{8\ \text{MHz}/16}{(51 + 1)} = 9615.38$$

Required Baud Rate = 9600

$$\text{Hence \% Error} = (9615.38 - 9600)/9600 = 0.16$$

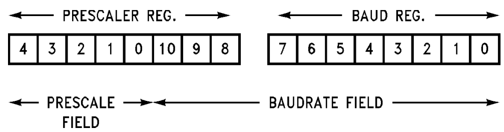
Which is well within the error limits.

The following is a sample assembly language routine illustrating BAUD Rate generation through PUT.

```
;This program will test the HPC16400E UART for 9600 baud.
;Using PUT for generating 9600 baud at 20 MHz
.sect code, rom 16
This is for 20 MHz CKI
;

;Using PUT for generating 9600 baud at 20 MHz
.sect code, rom 16
main:

                 ld 0x017e.w,#0x0000        ;for 9600 baud @ 20 MHz
                                            ;UDIV w/xtclk or xrclk (baud count)
                 ld 0x017c.w,#0x8033        ;baud div value to generate 9600 baud
                                            ;UDIVR (baud div) register
                 sbit 2, 0x122.b            ;xtclk
                 sbit 3, 0x122.b            ;xrclk

                 ld 0f2.b,#0x05             ;DIRB reg pin 1 outward direction.
                 ld 0f4.b,#0x05             ;BFUNL reg, turns on TDX bit

;char xmission
                 ld a,#041                  ;Load char "A"
xmit:
                 st a,0126.b                ;Load TBUF reg to transmit

                 jp xmit                    ;Continue to xmit

.endsect
.end main
```

3.0 BAUD RATE CALCULATIONS USING BRG (BAUD RATE GENERATOR).

The most flexible and accurate on-chip clocking is provided by the BAUD Rate generator and (BRG). The BAUD Rate generator is controlled by the register pair PSR and BAUD, shown below. The Prescale factor is selected by the upper 5 bits of the PSR register (the PRESCALE field), in units of the CK2 clock from 1 to 16 in ½ step increments. The lower 3 bits of the PSR register, in conjunction with the 8 bits of the baud register, form the 11-bit BAUDRATE field, which defines a baud rate divisor ranging from 1 to 2048, in units of the prescaled clock selected by the PRESCALE field. In Asynchrnous Mode, the resulting baud rate is $\frac{1}{16}$ of the clocking rate selected through the BRG circuit. The maximum baud rate generated using BRG is 625 kbaud.



TL/DD/11292–3

Formula:

$$\text{Required Baud Rate} = \frac{\text{CKI}}{32 * N * P}$$

where CK = Input Clock

N = Baud Rate Divisor

P = Prescaler Division Factor

Note: This calculation is for Asynchronous mode of UART operation.

Suppose we need 9600 Baud with given Clock i.e., CKI = 20 MHz

then

Required Baud Rate = 9600

CKI = 20 MHz

From formula stated earlier for required baud rate, we have

$$9600 = \frac{20 \text{ MHz}}{30 * N * P}$$

$$\rightarrow N * P = \frac{20 \times 10^6}{32 * 9600}$$

$N * P = 65.1$

or $N = 65.1/P$

P, which is a prescaler factor, should be selected from Table II in such a way that "N" should be close to an integer. Therefore substituting values of P in the table and calculating N we have the following table.

TABLE II

P Prescaler	N N = (65.104/P)
1	65.104
1.5	43.402
2	32.552
2.5	26.041
3	21.701
3.5	18.601
4	16.276
4.5	14.467
5	13.020
5.5	11.837
6	10.850
6.5	10.016
7	9.300
7.5	8.680
8	8.138
8.5	7.659
9	7.233
9.5	6.853
10	6.510
10.5	6.200
11	6.918
11.5	5.661
12	5.425
12.5	5.203
13	5.008 ← Value Closest to an Integer
13.5	4.822
14	4.650
14.5	4.489
15	4.340
15.5	4.200
16	4.069

Now choose N in such a way that it's closest to an integer. Obviously N = 5.008 is the closest to being an integer therefore, the value of P when N = 5.008 is 13

$\rightarrow$ P = 13 and N = 5

Now from the table "UART Prescaler Factors" select the binary "Prescale field" using the value of N derived above.

Percentage error of the Baud Rate produced is:

from the above table P = 13 and N = 5.008

$$\therefore \text{Baud Rate} = \frac{20\text{ MHz}}{32 \times N \times P}$$

$$\frac{20 \times 10^6}{32 \times 5.008 \times 12} = 9600.02$$

$$\% \text{ error} = (9600.02 - 9600)/9600$$
$$= 0.0002\%$$

Which is obviously negligible.

UART Prescaler Factors

Prescale Field (Binary)	Prescaler Factor
00000	(Compatibility Mode)
00001	1
00010	1.5
00011	2
00100	2.5
00101	3
00110	3.5
00111	4
01000	4.5
01001	5
01010	5.5
01011	6
01100	6.5
01101	7
01110	7.5
01111	8
10000	8.5
10001	9
10010	9.5
10011	10
10100	10.5
10101	11
10110	11.5
10111	12
11000	12.5
11010	13.5
11010	13.5
11011	14
11100	14.5
11101	15
11110	15.5
11111	16

in Binary format

P = 11001   (N − 1) = 100

Therefore Prescaler field is P = 11001 and baud rate divisor or baud rate field N = 100

Referring to BRG register format in page 7 we can combine 5 bits of P and 11 bits of baud rate field to load Prescaler bits (PSR) and Baud Rate generate bits (BRG) respectively.

PSR = 11001

Baud Rate field (N − 1) = 0 0 0 0 0 0 0 0 1 0 0

Combined value in binary format is

1 1 0 0   1 0 0 0   0 0 0 0   0 1 0 0

which in hex is

C   8   0   4

therefore load BRG register with C804.

The following is a sample assemble language routine illustrating BAUD Rate generation through BRG.

```
;Baud rate generation using BRG register
;BAUD RATE = CKI/(32 * N * P) where P = 5 bit prescalar value and N = 11 bit
;baud rate filed. For 9600 baud at 20 MHz  →  NP = 52.083 and so P = 13 and N = 4
;
;At 16 MHz crystal (CKI) for PSR use #0c8 and for BAUD use #07
;At 20 MHz crystal (CKI) for PSR use #0c8 and for BAUD use #04
;*************************************************************
.sect code, rom16
main:                                ;First exit compatibility mode
                                     ;by writing to PSR register
              ld 012a.b,#0c8         ;load prescalar i.e., PSR reg
              ld 012c.b,#04          ;load baudrate field i.e., BAUD at 20 MHz
              ld 0120.b,#000         ;8 bit data, space (0) parity in ENU register.
              ld 0122.b,#080         ;ENUI register, 2 stop bits
              ld 0f2.b,#01           ;DIRB register pin 1 outward direction
              ld 0f4.b,#01           ;BFUNL register, turns on TDX bit
;Loop to continuously xmit chars at specified baud rate.
xmit:         ld a,#041              ;load char "A"
              st a,0126.b            ;Load TBUF reg to transmit
              jp xmit                ;Continue to xmit.
.endsect
.end main
```

Performance Comparison of PUT and BRG Regarding Higher Baud Rate Generation.

Let's take a case where the required Baud rate is 625k baud at 20 MHz.

PUT:

$$\text{BAUDR} + 1 = \frac{\text{CK2/16}}{\text{Required Baud Rate}}$$

$$\text{Therefore BAUDR} + 1 = \frac{10 \times 10^6/16}{625 \times 10^3}$$

$$\text{BAUDR} + 1 = 0.1$$

$$\text{BAUDR} = -(0.9)$$

Therefore we see that, PUT can not be used to generate 625k baud. The limit on PUT is 208.3k baud.

BRG:

$$\text{Baud Rate Required} = \frac{\text{CKI}}{32 * N * P}$$

$$625\text{k} = \frac{20 \times 10^6}{32 * N * P}$$

$$N \times P = 1$$

$$N = 1 \qquad P = 1$$

i.e. Prescale field = 00001 N − 1 = 00000000000
i.e., 0000 1000 0000 0000 = 0 × 0800

Therefore load BRG register with 0x 0800 to generate 625k baud @ 20 MHz

Conclusion:

Thus we see that the clocking techniques on new generation HPCs are more accurate and very flexible. Generation of higher rates can be done with relative ease. We can also observe that, using newer UART clocking techniques the percentage error i.e., difference between the required baud rate and the actual baud rate produced goes down significantly.

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998