Forkortelser anvendt i dette dokument :

1. Lsb : Least Significant bit (Bit position 0)
2. Msb : Most Significant bit (Bit position 7, hvis det er en 8 bit størrelse.

Bit position 15, hvis det er en 16 bit størrelse.

Bit position 31, hvis det er en 32 bit størrelse.

Bit position 63, hvis det er en 64 bit størrelse).

Konverter mellem binære tal, decimale tal og hexa decimale tal

|  |  |  |
| --- | --- | --- |
| Dec | Bin | Hex |
| 90 | 0101 1010 | 0x5A |
| 1 | 0000 0001 | 0x01 |
| 130 | 1000 0010 | 0x82 |
| 170 | 1010 1010 | 0xAA |
| 32 | 0010 0000 | 0x20 |
| 229 | 1110 0101 | 0xE5 |
| 201 | 1100 1001 | 0xC9 |

Ofte sætter man bit eller nulstiller bit i et register, eks. Sæt bit 5 og 3 i PortB => PortB |= (1<<5) | (1<<3) . Dette kan oversættes til PortB = PortB | ((1<<5) | (1<<3)).

Sæt 1 bit i et 8 bit register eller variabel 0x3E

Sæt bit position 0:

0x3E | 0x01 = 0011 1110b or 0000 0001b = 0011 1111b = 0x3F,

løsning i tabel ville være:

Var1 = 0x3E Var2 = 0x01 Resultat = Var1 | Var2 = 0x3E | 0x01 = 0x3F

|  |  |  |  |
| --- | --- | --- | --- |
| Var1 | Var2 | Resultat i hex | Bit position der skal sættes |
| 0xF8 1111 1000 | 0x01 0000 0001 | 0xF9 | 0 0000 0001 |
| 0x98 | 0x02 | 0x9A | 1 0000 0010 |
| 129 | 0x04 | 0x85 | 2 0000 0100 |
| 1111 0011b | 0x08 | 0xFB | 3 0000 1000 |
| 0x77 | 0x10 | 0111 0111 = 0x77 | 4 0001 0000 |
| 160 | 0x20 | 0xA0 | 5 0010 0000 |
| 160 | 0x40 | 0xE0 | 6 0100 0000 |
| 0x00 | 0x80 | 0x80 | 7 1000 0000 |

Sæt flere bit i et 8 bit register eller variabel 0x3E

Sæt bit position 0 og bit position 6:

0x3E | 0x41 = 0011 1110b | 0100 0001b = 0111 1111b = 0x7F

|  |  |  |  |
| --- | --- | --- | --- |
| Var1 | Var2 | Resultat i hex | Bit position der skal sættes |
| 0x38 | 0x41 | 0x79 | 0 og 6 0100 0001 |
| 0x90 | 0x0A | 0x9A | 1 og 3 0000 1010 |
| 129 0x81 | 0x24 | 0xA5 | 2 og 5 0010 0100 |
| 1111 0011b 0xF3 | 0x0C | 0xFF | 2 og 3 0000 1100 |
| 0100 0111 0x47 | 0x30 | 0111 0111 = 0x77 | 4 og 5 0011 0000 |
| 160 0xA0 | 0x03 | 0xA3 | 0 og 1 0000 0011 |
| 160 0xA0 | 0x4C | 0xEC | 2, 3 og 6 0100 1100 |
| 0x00 | 0xA6 | 0xA6 | 1 og 2 og 5 og 7  1010 0110 |

Clear 1 bit i et 8 bit register eller variabel 0x3F nulstil bit position 0.

0x3F & 0xFE = 0011 1111b & 1111 1110b = 0011 1110b = 0x3E

|  |  |  |  |
| --- | --- | --- | --- |
| Var1 | Var2 | Resultat i hex | Bit position der skal nulstilles |
| 0xC7 | 1111 1110 | 0xC6 | 0 |
| 0x8A | 1111 1101 | 0x88 | 1 |
| 0x55 | 1111 1011 | 0x51 | 2 |
| 1111 1011b | 1111 0111 | 0xF3 | 3 |
| 0x37 | 1110 1111 | 0010 0111 = 0x27 | 4 |
| 160 | 1101 1111 0xDF | 0x80 | 5 |
| 160 | 1011 1111 0xBF | 0xA0 | 6 |
| 0x81 | 0111 1111 0x7F | 0x01 | 7 |

Clear flere bit i et 8 bit register eller variabel nulstil bit position 0 og bit position 3.

0x3F & 0xF6 = 0011 1111b & 1111 0110b = 0011 0110b = 0x36

|  |  |  |  |
| --- | --- | --- | --- |
| Var1 | Var2 | Resultat i hex | Bit position der skal nulstilles |
| 0xC7 | 01111110 | 0x46 | 0 og 7 |
| 0x88 | 11011101 0xDD | 0x88 | 1 og 5 |
| 0x55 | 11101011 | 0x41 | 2 og 4 |
| 1111 1011b | 10110111 | 0xb3 | 3 og 6 |
| 0010 0111 = 0x27 | 01101111 | 0010 0111 = 0x27 | 4 og 7 |
| 160 | 01011111 | 0x00 | 5 og 7 |
| 163 | 01111100 | 0x20 | 0 og 1 og 7 |
| 0x87 | 01111000 | 0x00 | 0 og 1 og 2 og 7 |
|  |  |  |  |

Den Sjove Opgave !!! :

Hans sætter 760kr i Embedded Programmerings banken i Logikland. I løbet af det første år Hans’ penge står i banken, sker der følgende transaktioner med det indestående beløb på Hans’ konto (det skal lige oplyses at alle indestående beløb på de forskellige konti og hermed også Hans’ konto er indeholdt i en 16 bit variabel. Dette skal I huske på, når I laver jeres beregninger !!!). Husk også at regne i det/de rigtige talsystemer, Ellers kan I ikke regne opgaven.

Sluttelig husk at vise alle jeres beregninger. Når det er noget med penge, skal dokumentationen være i orden !!!

1. Hans’ penge udsættes for en **bitvis OR** med Daniels penge. Her skal det oplyses, at Daniel har et beløb på 1586kr stående i banken.  
     
   Hans: 0b01011111000  
   Daniel: 0b11000110010  
   Res: 0b11011111010
2. Det nye beløb udsættes nu for en **bitvis EXCLUSIVE OR** med Mihealas indestående i banken som er på 17185kr.  
     
   uint16\_t num1 = 1786;

uint16\_t num2 = 17185;  
uint16\_t num3 = 17883;

1. Der laves nu en **logisk OR** med Emils penge. Emil har et indestående i banken på 4660kr.  
     
   uint16\_t num1 = 17883;

uint16\_t num2 = 4660;

uint16\_t num3 = 1;

1. Herefter laves der en **bitvis AND** med Lasses beløb i banken. Lasse indestående er på 13398kr  
     
   uint16\_t num1 = 1;

uint16\_t num2 = 13398;

uint16\_t num3 = 0;

1. Det nye beløb negeres nu.  
     
   uint16\_t num3 = 65535;
2. Herefter laves der en **logisk AND** med Rasmus’ beløb i banken. Rasmus har et indestående beløb på 17767kr.   
     
   uint16\_t num1 = 65535;

uint16\_t num2 = 17767;

uint16\_t num3 = 1;

1. På det nye beløb laves en **bitvis OR** med Niels’ indestående i banken som er på 22136kr.  
     
   uint16\_t num1 = 1;

uint16\_t num2 = 22136;

uint16\_t num3 = 22137;

1. Sluttelig laves der en **bitvis AND** med Omads indestående i banken. Omad har et indestående i banken på 61680kr.  
     
   uint16\_t num1 = 22137;

uint16\_t num2 = 61680;

uint16\_t num3 = 20592;

Hvor meget har Hans i banken efter de ovennævnte operationer ???

uint16\_t num3 = 20592;