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 | 0x01 | 0xF9 | 0 |
| 0x98 | 0x02 | 0x9A | 1 |
| 129 | 0x03 | 0x89 | 2 |
| 1111 0011b | 0x04 | 0xFB | 3 |
| 0x87 | 0x05 | 0111 0111 = 0x77 | 4 |
| 160 | 0x06 | 0xA0 | 5 |
| 160 | 0x07 | 0xE0 | 6 |
| 0x00 | 0x08 | 0x80 | 7 |

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 | 0xE9 | 0 og 6 |
| 0x90 | 0x0A | 0x9A | 1 og 3 |
| 129 | 0x24 | 0x95 | 2 og 5 |
| 1111 0011b | 0x0C | 0xFF | 2 og 3 |
| 0x47 | 0x30 | 0111 0111 = 0x77 | 4 og 5 |
| 160 | 0x03 | 0xA3 | 0 og 1 |
| 160 | 0x4C | 0xEC | 2, 3 og 6 |
| 0x00 | 0xA6 | 0xA6 | 1 og 2 og 5 og 7 |

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 | 0xFE | 0xC6 | 0 |
| 0x88 | 0xFD | 0x88 | 1 |
| 0x55 | 0xFC | 0x51 | 2 |
| 1111 1011b | 0xFB | 0xF3 | 3 |
| 0x37 || 0x27 | 0xFA | 0010 0111 = 0x27 | 4 |
| 160 | 0xF9 | 0x80 | 5 |
| 160 | 0xF8 | 0xA0 | 6 |
| 0x81 | 0xF7 | 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 | 0xEE | 0x46 | 0 og 7 |
| 0x88 | 0xDD | 0x88 | 1 og 5 |
| 0x55 | 0xEB | 0x41 | 2 og 4 |
| 1111 1011b | 0xBE | 0xB3 | 3 og 6 |
| 0x27 | 0x6F | 0010 0111 = 0x27 | 4 og 7 |
| 160 | 0x5F | 0x40 | 5 og 7 |
| 163 | 0x7C | 0x40 | 0 og 1 og 7 |
| 0x87 | 0x78 | 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: 760kr  Daniel: 1586kr  Hans binær: 0000 0001 0111 1100  Daniel binær: 0000 0110 0011 0010  Resultat: 0000 0111 0111 1110  1918kr = 1024 + 512 + 256 + 64 + 32 + 16 + 8 + 4 + 2 |

1. Det nye beløb udsættes nu for en **bitvis EXCLUSIVE OR** med Mihealas indestående i banken som er på 17185kr.

|  |
| --- |
| Hans: 1918kr  Mihaela: 17.185kr  Hans binær: 0000 0111 0111 1110  Mihaela binær: 0100 0011 0010 0001  Resultat: 0100 0100 0101 1111  17.503kr = 16384 + 1024 + 64 + 16 + 8 + 4 + 2 +1 |

1. Der laves nu en **logisk OR** med Emils penge. Emil har et indestående i banken på 4660kr.

|  |
| --- |
| Hans: 17.503kr  Emil: 4660kr  Resultat = 1kr |

1. Herefter laves der en **bitvis AND** med Lasses beløb i banken. Lasse indestående er på 13398kr

|  |
| --- |
| Hans: 1kr  Lasse: 13.398kr  Hans binær:  0000 0000 0000 0001  Lasse binær: 0011 0100 0101 0110  Resultat: 0kr |

1. Det nye beløb negeres nu.

|  |
| --- |
| Hans: 0kr  Hans binær: 1111 1111 1111 1111  Resultat: 65.535kr |

1. Herefter laves der en **logisk AND** med Rasmus’ beløb i banken. Rasmus har et indestående beløb på 17767kr.

|  |
| --- |
| Hans: 65.535kr  Rasmus: 17.767kr  Resultat: 1kr |

1. På det nye beløb laves en **bitvis OR** med Niels’ indestående i banken som er på 22136kr.

|  |
| --- |
| Hans: 1kr  Niels: 22.136kr  Hans binær: 0000 0000 0000 0001  Niels binær: 0101 0110 0111 1000  Resultat: 0101 0110 0111 1001  22.137kr = 16384 + 4096 + 1024 + 512 + 64 + 32 + 16 + 1 |

1. Sluttelig laves der en **bitvis AND** med Omids indestående i banken. Omid har et indestående i banken på 61680kr.

|  |
| --- |
| Hans: 22.137kr  Omid: 61.680kr  Hans binær: 0101 0110 0111 1001  Omid binær: 1111 0000 1111 0000  Resultat: 0101 0000 0111 0000  20.592kr = 16384 + 4096 + 64 + 32 + 16 |

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