

```
//
typedef unsigned int UINT;

////////////////////////////////////
// FPGA Modules

//*****
// Hex display

//Hex display value
//Bits 7..4 represent the left hex display
//Bits 3..0 represent the right hex display
UINT* const g_cpwHEXVAL = (UINT*)(0x0A000000);

//Hex control register
//Bit 0 controls the blanking input
UINT* const g_cpwHEXCTL = (UINT*)(0x0A000002);

//*****
//FPGA control

//FPGA control
UINT* const g_cpwFPGA = (UINT*)(0x08000001);

//Upper Level Control / Register value
UINT* const g_cpwULCTL = (UINT*)(0x14000000);

//*****
//Dip Switches

//Dip switch value
UINT* const g_cpwDIPSW = (UINT*)(0x04000000);

//*****
//Digital to Analog Converter

//DAC control register (controls reset and MSB)
UINT* const g_cpwDACctl = (UINT*)(0x02000000);

//Holds the value of DAC channel A
UINT* const g_cpwDACChA = (UINT*)(0x02000002);

//Holds the value of DAC channel B
UINT* const g_cpwDACChB = (UINT*)(0x02000004);

UINT* const g_cpwPMC = (UINT*)(0x12000000);
UINT* const g_cpwPESNET = (UINT*)(0x10000000);
UINT* const g_cpwDSP = (UINT*)(0x16000000);

//*****
//Peripheral port

//Peripheral control register.
//Set to 0x00000000 for DSP control of peripheral port data
UINT* const g_cpwPeriphCtl = (UINT*)(0x0C000000);

//Peripheral port A output data control (low side)
//Pins 32..1 map to bits 31..0
UINT* const g_cpwPeriphAVal_L = (UINT*)(0x0C000002);

//Peripheral port A output data control (high side)
//Pins 39..33 map to bits 6..0
UINT* const g_cpwPeriphAVal_H = (UINT*)(0x0C000004);

//Peripheral port A direction control (low side)
//Pins 32..1 map to bits 31..0
//Set to 1 for that pin to be an output. Set to zero for that pin to be an input.
UINT* const g_cpwPeriphADir_L = (UINT*)(0x0C000022);

//Peripheral port A direction control (high side)
```

```
//Pins 39..33 map to bits 6...0
UINT* const g_cpwPeriphADir_H = (UINT*) (0x0C000024);

//Peripheral port D value control (low side)
//Pins 32..1 map to bits 31..0
UINT* const g_cpwPeriphDVal_L = (UINT*) (0x0C000006);

//Peripheral port D value control (high side)
//Pins 48..33 map to bits 15..0
UINT* const g_cpwPeriphDVal_H = (UINT*) (0x0C000008);

//Peripheral port D direction control (low side)
//Pins 32..1 map to bits 31..0
//Set to 1 for that pin to be an output. Set to zero for that pin to be an input.
UINT* const g_cpwPeriphDDir_L = (UINT*) (0x0C000026); //1 = output, 0 = input

//Peripheral port D direction control (high side)
//Pins 48..33 map to bits 15..0
//Set to 1 for that pin to be an output. Set to zero for that pin to be an input.
UINT* const g_cpwPeriphDDir_H = (UINT*) (0x0C000028);

////////////////////////////////////
// Functions

void HEX_WRITE( unsigned char chVal );
void HEX_BLANK( bool bBlank );

unsigned int DIP_GETVAL();
void PREP_FPGA();
void DAC_SETVAL( UINT nCha, UINT nChb, unsigned char nCH);

//Used to prepare the FPGA for reading or writing. Needed as a workaround since
//we cannot read or write the same address two times in a row.
#define PREP_FPGA() *g_cpwFPGA = 0x00000000

inline unsigned int* FORM_PTR( unsigned int* p1, unsigned int* p2 )
{
    return (unsigned int*)( (unsigned int)(p1) | (unsigned int)(p2) );
}
```