```
#define BME680_H1_MSB_REG   (27)
#define BME680_H3_REG       (28)
#define BME680_H4_REG       (29)
#define BME680_H5_REG       (30)
#define BME680_H6_REG       (31)
#define BME680_H7_REG       (32)
#define BME680_T1_LSB_REG   (33)
#define BME680_T1_MSB_REG   (34)
#define BME680_GH2_LSB_REG  (35)
#define BME680_GH2_MSB_REG  (36)
#define BME680_GH1_REG      (37)
#define BME680_GH3_REG      (38)

/** BME680 register buffer index settings*/
#define BME680_REG_FILTER_INDEX     UINT8_C(5)
#define BME680_REG_TEMP_INDEX       UINT8_C(4)
#define BME680_REG_PRES_INDEX       UINT8_C(4)
#define BME680_REG_HUM_INDEX        UINT8_C(2)
#define BME680_REG_NBCONV_INDEX     UINT8_C(1)
#define BME680_REG_RUN_GAS_INDEX    UINT8_C(1)
#define BME680_REG_HCTRL_INDEX      UINT8_C(0)

/** BME680 pressure calculation macros */
/*! This max value is used to provide precedence to multiplication or division
 * in pressure compensation equation to achieve least loss of precision and
 * avoiding overflows.
 * i.e Comparing value, BME680_MAX_OVERFLOW_VAL = INT32_C(1 << 30)
 */
#define BME680_MAX_OVERFLOW_VAL      INT32_C(0x40000000)

/** Macro to combine two 8 bit data's to form a 16 bit data */
#define BME680_CONCAT_BYTES(msb, lb)   (((uint16_t)msb << 8) | (uint16_t)lsb)

/** Macro to SET and GET BITS of a register */
#define BME680_SET_BITS(reg_data, bitname, data) \
        ((reg_data & ~(bitname##_MSK)) | \
        ((data << bitname##_POS) & bitname##_MSK))
```