Integer types

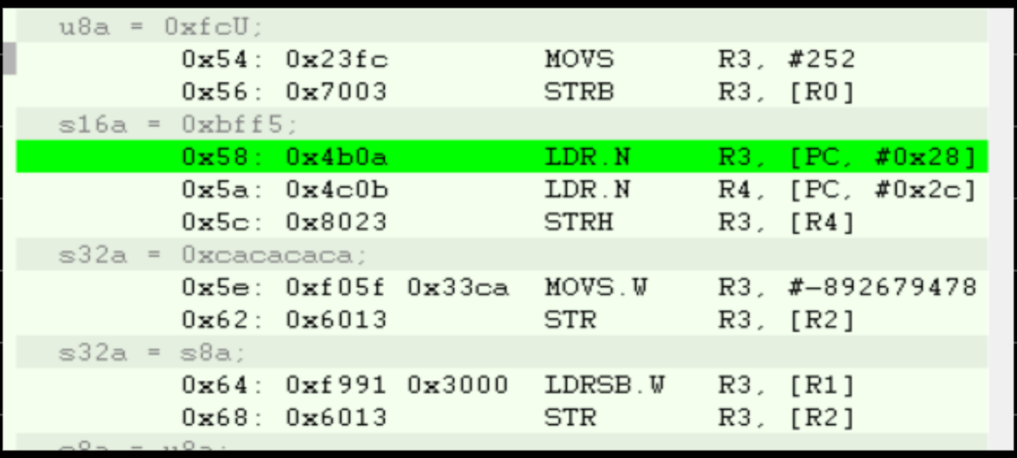
In C size of int, char, unsigned is generally not set and is adjusted according to the processor. Int can be 8 bit, 16 bit or above based on targeted processor running the C code.
Thus to define standards in variable type sizes, we use #include<stdint.h> header file.

Code:

```
#include "tm4c123gh6pm.h"
#include <stdint.h>

int8_t s8a;
int16_t s16a;
int32_t s32a;
uint8_t u8a;

int main()
{

  // Checking sizeof types
  u8a = sizeof(uint8_t);
  s8a = sizeof(int8_t);
  s32a = sizeof(int32_t);

  // Assigning values to variables
  u8a = 0xfcU;
  s16a = 0xbff5;
  s32a = 0xcacacaca;


  s32a = s8a;
  s8a = u8a;

  return 0;
}
```

The variables are assigned the values using MOVS/MOVS W

The variables of different sizes are also stored using slightly different instructions. STRB for 8 bit variables, STRH for 16 bits and, STR for 32 bit variables:



```
    u8a = 0xfcU;
              0x54: 0x23fc           MOVS       R3, #252
              0x56: 0x7003           STRB       R3, [R0]
    s16a = 0xbff5;
              0x58: 0x4b0a           LDR.N      R3, [PC, #0x28]
              0x5a: 0x4c0b           LDR.N      R4, [PC, #0x2c]
              0x5c: 0x8023           STRH       R3, [R4]
    s32a = 0xcacacaca;
              0x5e: 0xf05f 0x33ca    MOVS.W     R3, #-892679478
              0x62: 0x6013           STR        R3, [R2]
    s32a = s8a;
              0x64: 0xf991 0x3000    LDRSB.W    R3, [R1]
              0x68: 0x6013           STR        R3, [R2]
```