UG1047 June 4, 2014

# LibXil RSA for Zynq-7000 AP SoC Devices (v1.1)

## LibXil RSA Library Overview

The LibXil RSA library provides APIs to use RSA encryption and decryption algorithms and SHA algorithms.

For an example on usage of this library, refer to the RSA Authentication application and its documentation.

## SDK Project Files and Folders

The following table lists and describes the LibXil RSA files and folders.

*Table 1-1:* **LibXil RSA Files and Folders**

| File/Folder | Description |
|---|---|
| librsa.a | Contains the implementation |
| xilrsa.h | Header containing APIs. |

## LibXil RSA Library Description

The `xilrsa` library contains the description of the RSA and SHA functions that you use to create and verify the signature. The RSA-2048 bit is used for RSA and the SHA-256 bit is used for hash.

### Use of SHA-256 functions

When all the data is available on which `sha2` must be calculated, the `sha_256()` can be used with appropriate parameters, as described.

When all the data is not available on which `sha2` must be calculated, use the `sha2` functions in the following order:

1. `sha2_update()` can be called multiple times till input data is completed.
2. `sha2_context` is updated by the library only; do not change the values of the context.

### SHA2 Example

```
sha2_context ctx;
sha2_starts(&ctx);
sha2_update(&ctx, (unsigned char *)in, size);
sha2_finish(&ctx, out);
```

### Class

```
struct sha2_context
```

## Macro

### RSA definitions

```
#define RSA_DIGIT unsigned long
#define RSA_NUMBER[1] RSA_DIGIT
```

1. RSA_NUMBER is a pointer to RSA_DIGIT

## LibXil RSA APIs and Descriptions

```
void rsa2048_exp (const unsigned char *base, const
   unsigned char *modular, const unsigned char
   * modular_ext, const unsigned char *exponent, unsigned
   char *result)
```

| | |
|---|---|
| Parameters | modular: a char pointer which contains the key modulus<br>modular_ext: a char pointer which contains the key modulus extension<br>exponent: a char pointer which contains the private key exponent<br>result: a char pointer which contains the encrypted data |
| Returns | None |
| Description | This function is used to encrypt the data using 2048 bit private key. |
| Includes | xilrsa.h |

```
void rsa2048_pubexp (RSA_NUMBER a, RSA_NUMBER x, unsigned
   long e, RSA_NUMBER m, RSA_NUMBER rrm )
```

| | |
|---|---|
| Parameters | a: RSA_NUMBER containing the decrypted data.<br>x: RSA_NUMBER containing the input data<br>e: unsigned number containing the public key exponent<br>m: RSA_NUMBER containing the public key modulus<br>rrm: RSA_NUMBER containing the public key modulus extension. |
| Returns | None |
| Description | This function is used to decrypt the data using 2048 bit public key |
| Includes | xilrsa.h |

---

```
void sha2_finish (sha2_context * ctx, unsigned char
   * output )
```

| | |
|---|---|
| Parameters | ctx: Pointer to sha2_context structure.<br>output: char pointer to calculated hash data. |
| Returns | None |
| Description | This function finishes the SHA calculation. |
| Includes | xilsha.h |

---

```
void sha2_starts (sha2_context *  ctx)
```

| | |
|---|---|
| Parameters | ctx: Pointer to sha2_context structure that stores status and buffer. |
| Returns | None |
| Description | This function initializes the sha2 context. |
| Includes | xilsha.h |

```
void sha2_update (sha2_context * ctx, unsigned char
   * input, unsigned int ilen )
```

| | |
|---|---|
| Parameters | ctx: Pointer to `sha2_context`, structure.<br>input: Char pointer to data to add.<br>ilen: Length of the data. |
| Returns | None |
| Description | This function adds the input data to SHA-256 calculation. |
| Includes | xilsha.h |

---

```
void sha_256 (const unsigned char * in, const unsigned
   int size, unsigned char * out)
```

| | |
|---|---|
| Parameters | in: Char pointer which contains the input data.<br>size: Unsigned int which contains the length of the input data.<br>out: Output buffer that contains the hash of the input. |
| Returns | None |
| Description | This function calculates the hash for the input data using SHA-256 algorithm. This function internally calls the `sha2_init`, updates and finishes functions and updates the result. |
| Includes | xilrsa.h |