AES ECB uses EasyDMA with scatter-gather to access to memory for in-place operations on cleartext and ciphertext during encryption. ECB uses the same AES core as the CCM and AAR blocks and is an asynchronous operation which may not complete if the AES core is busy.

AES ECB performs a 128 bit AES block encrypt. At the START task, cleartext is loaded into the ECB from memory described by the scatter/gather job list pointed to by INPTR and the ciphertext is written into memory described by the job list pointed to by OUTPTR. When the last cleartext byte has been encrypted and written to OUTPTR, the END event is triggered.

The following figure illustrates how the input and output job lists can be configured. For more details of the joblists, see EasyDMA on page 264.

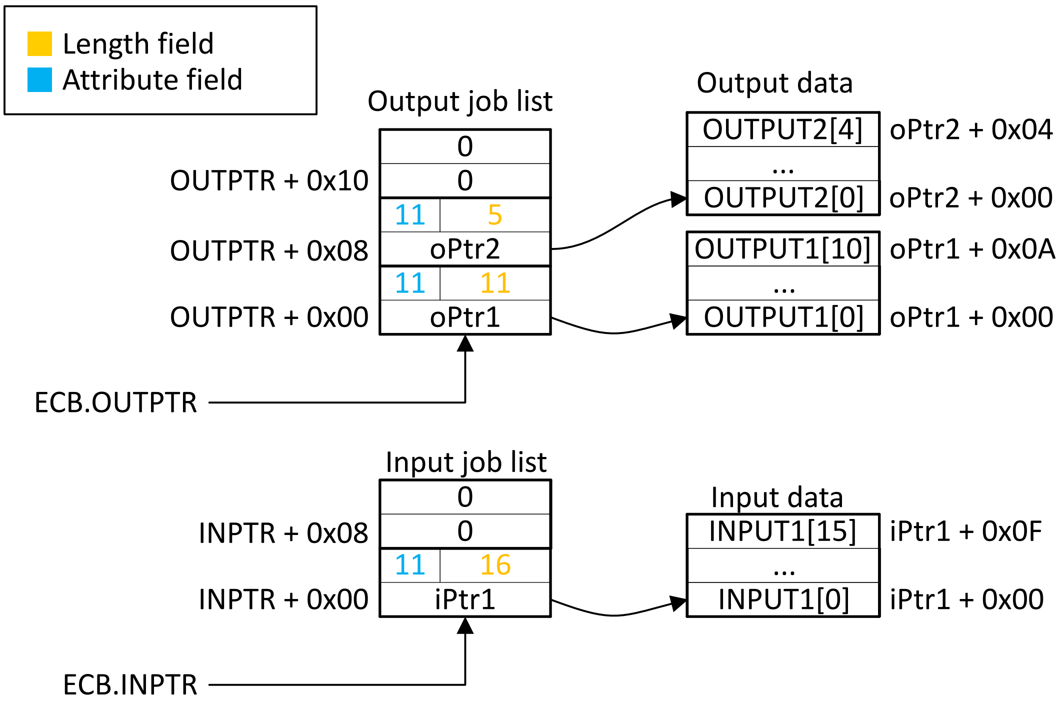

Figure 58: Example job lists for ECB operation

The AES key is set by writing the KEY.VALUE key registers. The same key can be used to encrypt multiple blocks by triggering the START task multiple times.

AES ECB can be stopped by triggering the STOP task.

ECB only supports a single 16-byte block. For different job list sizes the following rules apply:

- If less than 16 bytes is supplied as input, then a PrematureInptrEnd error is triggered. The EVENTS_ERROR will also be set.
- If more than 16 bytes is supplied as input, then only the first 16 bytes are used.
- For an output job, only the number of bytes specified in the job list are copied to memory.

The 128-bit key in the KEY.VALUE registers is stored in reverse byte order relative to the payload. For example, using the sample calculation from the Bluetooth Core Specification v5.4, Volume 6, Part C, chapter 1.1, with the following data:

- Key: 4C68384139F574D836BCF34E9DFB01BF
- Plaintext: 0213243546576879acbdcedfe0f10213
- Expected Encrypted Output: 99ad1b5226a37e3e058e3b8e27c2c666

The KEY.VALUE registers are populated as follows:

- KEY.VALUE[0] = `0x9DFB01BF`
- KEY.VALUE[1] = `0x36BCF34E`