Disclaimer: This document describes the current design philosophy
Details may changes as the design evolves.

Image File format

The table below describes the general format. More detail is provided with extracts from an example file at the end of this document.

Description	Note	Byte length	File Address
File Identifier	*1*	8	0000 Bytes
Protocol Version number	*2*	2	
Number of Images in Bank (n)	*3*	2	
Timing Function register (TFR)	*9,13*	2	
Reserved area		50	
		Start Index Field at this boundary	
Image 0, Index	*4,5*	1	0064 Bytes
Image 0, Length in Points (m)	*6*	3	
Image 0, Memory Offset Address (bytes)	*7*	4	
Image 1, Index		1	
Image 1, Length in Points (p)		3	
Image 1, Memory Offset Address (bytes)		4	
🡻			
Image (n), Index		1	
Image (n), Length in Points		3	
Image (n), Memory Offset Address	*8*	4	
Image Sequence Table (IST)	*10,13*	8, per image output	
		Start Image Data at this boundary	
Image 0, Header (Memory Offset Address 0 h)	*11*	16	4096 Bytes
Image 0, Point 1	*12*	16, per channel	
Image 0, Point 2		16, per channel	
🡻		"	
Image 0, Point (m)		16, per channel	
Image 1, Header		16	
Image 1, Point 1		16, per channel	
Image 1, Point 2		16, per channel	
🡻		"	
Image 1, Point (p)		16, per channel	
……. repeat for (n) images ……			

Table notes:

1: ASCII coded string for ' iHHS_IMG ' = 694848535F494D47 h
where _IMG indicates an Image file

2: Protocol version = 0001 h

3: Limit = 254 images per memory bank (00FE h)

4: Start of the Index field at address 64 bytes (40 h)

5: Image Index number must be unique for each image

6: Image Length expressed in image points (m)
(where each point is a new output across all enabled DDS channels)
Minimum: 2
Maximum: 1,000,000 (4-channel output)
2,000,000 (2-channel output)

7: Start address for the Image in memory space
i.e. it is the Memory Offset not the File address.

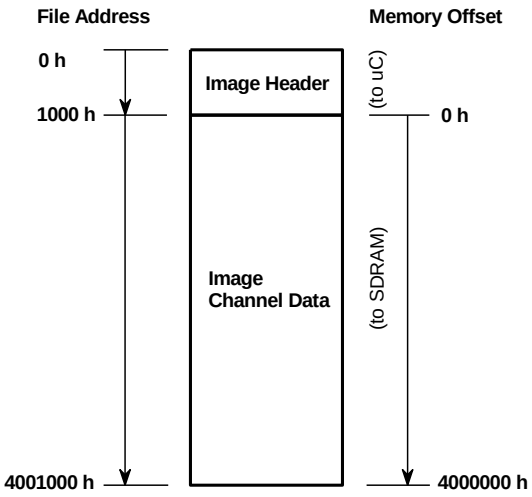
First available Memory Offset address: 0
Last available Memory Offset address: 64 Mbytes

Offset for Image "n" = Offset Image "n-1" + total byte length for Image "n-1"

where:

total byte length
= (1*image header + m*image data points x channels per point) x 16

channels per point = 1 to 4



8: End of Index field

9: Timing Function Register (TFR)

Defines the point clock source and the trigger method for each image start.

TFR	Point Clock	Image Trigger	Note
0000	INT_OSC	IMG_DLY	Image trigger delay defined in IST table (note 9)
0001	INT_OSC	EXT_TRG	
0002	INT_OSC	INT_TRG	INT = Internally generated/software initiated
0003	INT_OSC	CONT	CONT = no triggers required
0004	EXT_CLK	IMG_DLY	
0005	EXT_CLK	EXT_TRG	EXT = Hardware input from an external source
0006	EXT_CLK	INT_TRG	
0007	EXT_CLK	CONT	

10: Image Sequence Table (IST)

Contains the clock, trigger and delay parameters for the image referenced by the index number. Images can be repeated with the same or different parameters.
This is located immediately after the last entry in the list of Image Index's

INDEX: unique reference number for each image downloaded into the memory bank.
Up to 64 images can be downloaded. The special INDEX value of FF h is the IST termination byte and indicates the end of the sequence. The subsequent byte(s) define the final action.
The termination byte is always required as the last INDEX entry line of the IST.

SCR_DLY: delay added to specified bits in the synchronous digital output.

INT_OSC: Period of internally generated point clock. (ref: TFR table)
or
CLK_DIV: external clock input divider
(Function defined by TFR clock selection INT_OSC or EXT_CLK)

IMG_DLY: Delay between the end of this Image and the start of next. (ref: TFR table)

REPEAT: Repeat current image with same parameters up to 255 times within the sequence.
Note: base 0 numbered thus:

1x play of current image, REPEAT = 0
2x play of current image, REPEAT = 1

IST Parameter	INDEX	REPEAT	SCR_DLY	INT_OSC (CLK_DIV)*	IMG_DLY	Total Bytes
Range	0-254	0-255	0-6.5msec	1usec-65msec (1 – 65K)*	0 – 6.5sec	
Resolution	1	1	0.1usec	1usec (Ext_Clk period)*	0.1msec	
Length(bytes)	1	1	2	2	2	8
Min	00 h	00 h	0000 h	0001 h	0000 h	
Max	3F h	FF h	FFFF h	FFFF h	FFFF h	
End of IST	FF h	Refer table below				

Each line entry of the table is 8 bytes long ("Total")
Min table length: 1 line
Max table length: 512 lines

* Function defined by TFR clock selection INT_OSC or EXT_CLK

End of sequence table

The end of sequence is identified by the termination byte in the INDEX field. The subsequent bytes define the final action.

All repeat actions are indefinite. A 'Force Stop' instruction will be required to end output.

End of Table	INDEX	Action		INDEX Pointer	RFU	Combined
Length(bytes)	1	1		2	4	8
		Value	*Description*	0 - 63		
	FF h	00	*STOP*	XXXX	XXXXXXXX	
	FF h	01	*Repeat entire sequence*	XXXX	XXXXXXXX	
	FF h	02	*Repeat sequence starting at INDEX pointer*	00nn h	XXXXXXXX	
	FF h	03	*Repeat a specific Image identified by INDEX pointer*	00nn h	XXXXXXXX	

The Force stop bit is implemented via the FPGA OpControl register.
The firmware writes a 16 bit word sent from the PC App to the FPGA OpControl register.
Bit 1 is the Force Stop control bit and this is specified as stopping the output.
Bit 0 could also be used to stop at the end of the current image.

11: **Header data**, one per image, 16 bytes total

Note	Description	Comment	Register Address and Contents			
a	**HSCR_0**	2 channels per	**0x4**	**2**		
a	**HSCR_1**	All SDOR Bits delayed	**0x5 *(or 6)***	**FFF**		
			(0x5 = level , 0x6 = pulsed)			
	CSR	All channels selected	**0x00**	**F4**		
	FR1		**0x01**	**D0**	**00**	**00**
	FR2		**0x02**	**20**	**00**	
	CFR		**0x03**	**00**	**03**	**00**

Key:

Address
Data

Note a:

Header Sync Control Registers (HSCR_n)

Description		**Output type**	**Action**	R/-W, addr(3)	_ _ _ _	_ _ _ _	_ _ _ _
SCR_Hr0	**# channels per point**	*na*	*na*	**0100**	*d3..d0*	*na*	*na*
SCR_Hr1	**Level SCR delay mask**	**Level**	**1= delay enable**	**0101**	*d11..d8*	*d7..d4*	*d3..d0*
SCR_Hr1	**Pulse SCR delay mask**	**Pulsed**	**1= delay enable**	**0110**	*d11..d8*	*d7..d4*	*d3..d0*

Under most operating modes, only the 2 Header Sync Control registers (HSCR) would be changed by the user application. All points in the corresponding image are affected by the settings in the HSCR_0 and HSCR_1.

The channel select register CSR, function registers FR1, FR2 and channel function register CFR set up the correct operating mode for the DDS chip. These would not normally be changed by the user application. The correct values are shown.

HSCR_0
HSCR_0 defines the number of active outputs per image point. This can change from image to image.

For 2 channels per image point, HSCR_0 = 42 h
For 4 channels per image point, HSCR_0 = 44 h

HSCR_1
Each point on the image can be assigned one of four further sync control register (see Channel Sync Control Register below). HSCR_1 provides additional functionality to the 12-bit SDOR option (only) of the Channel Sync Control register.

Synchronous digital output register (SDOR) output type is defined by the HSCR_1 register address.

- For the SDOR bits to be an output *level*, use address *05h* for HSCR_1
The levels can be programmed to change once per point update clock.

- For the SDOR bits to be a *pulsed* output level, use address *06h* for HSCR_1
Pulses are output once per point update clock.
The pulse period is 50% of the INT_OSC value (or 1usec if the EXT_CLK input is selected).

In both cases, one or more of these outputs may be delayed by the value set in the IST table (note 10), provided the appropriate mask bit is set in HSCR_1. This mask is the 12-bit data that follows the 05h or 06h address pointer.

	Level output	Pulse output
No bits delayed:	HSCR_1 = 5000 h	HSCR_1 = 6000 h
Bit 4 and 11 delayed:	HSCR_1 = 5810 h	HSCR_1 = 6810 h
All bits delayed:	HSCR_1 = 5FFF h	HSCR_1 = 6FFF h

12: **Channel data**, 2 (or 4) sets per image point. Each set = 16 bytes

Note	Description	Comment	Register Address and Contents				
Point 1		(2 channels per point)					
B	**CSCR_0**	SDOR Bits all high	**0x4**	**FFF**			
	CSR	Channel 0	**0x00**	**14**			
	CFTW0 (Freq)	51.00 MHz	**0x04**	**1A**	**1C**	**AC**	**08**
	CPOW0 (Phase)	180 deg	**0x05**	**20**	**00**		
	ACR (Amplitude)	100%	**0x06**	**00**	**13**	**FF**	
B	**CSCR_1**		**0x3**	**000**			
	CSR (channel1)	NOP	**0x00**	**24**			
	CFTW0 (Freq)	67.67 MHz	**0x04**	**22**	**A5**	**A4**	**69**
	CPOW0 (Phase)	0 deg	**0x05**	**00**	**00**		
	ACR (Amplitude)	70%	**0x06**	**00**	**12**	**CC**	
Point 2	**repeat n points**						

Key:

Address byte	**Address**
Data byte	**Data**

Note b:

Channel Sync Control Register (CSCR_n)

This provides a choice of functions. There are 5 options. Each channel of each point can feature a different function.

CSCR_n Item Description		R/-W [3], Addr [2:0]	*data*	*data*	*data*
NOP		**0011**	*x x x x*	*x x x x*	*x x x x*
Sync Digital Output register	**(SDOR)**	**0100**	*d11..d8*	*d7..d4*	*d3..d0*
Sync DAC_0 data	**(SDAC0)**	**0101**	*d11..d8*	*d7..d4*	*d3..d0*
Sync DAC_1 data	**(SDAC1)**	**0110**	*d11..d8*	*d7..d4*	*d3..d0*
TBD		**0111**	*d11..d8*	*d7..d4*	*d3..d0*

(shown for write to register i.e. bit-3 of the address nibble = 0)

The options available depend on the number of channels per point.

For a configuration with 2 channels per image point, each point could have two specific channel sync control functions. e.g. Digital Output Register SDOR and SDAC_0

The next point in the image could have the same or different choice of channel sync control registers.

Sync Digital Output (SDOR)
All 12 bits are freely programmable and 8 (minimum) will be available to the outside world. The HSCR_1 register pointer in the Image Header Data defines if these bits are a level or pulsed output.

(Ref: iDDS-2F; It is no longer required for the user to program an equivalent TACHO bit in a sync control register in order to monitor progress through the image. Specific registers will track progress through the image sequence table and a count the current location within that active image. This data can be requested via USB at any time.)

Individual bits of the SDOR are delayed by the value set in the IST table (note 10), provided the appropriate mask bit is set in SCR_Hr1 (note 11a).

All bits low: SDOR = 4000 h
Only bits 3 and 10 high: SDOR = 4408 h
All bits high: SDOR = 4FFF h

Sync DAC Outputs (SDAC_0 and SDAC_1)
Two independent analog outputs are available to the outside world, each with 12-bit resolution. Range 0 - 5V

0V: SDAC_0 = 5000 h or SDAC_1 = 6000 h
2.5V: SDAC_0 = 57FF h or SDAC_1 = 67FF h
5V: SDAC_0 = 5FFF h or SDAC_1 = 6FFF h

NOP = 3000 h
Setting when no Sync control data is required

DDS registers

All the channel registers need be programmed with a valid entry including Channel Select Register CSR, Channel Frequency Tuning Word CFTW0, Channel Phase Offset Word CPOW0 and the Amplitude Control Register ACR.

Channel Select Register

The 8-bit CSR selects the channel to be programmed and the chip comms' mode. The mode is fixed. It is determined by the lower 4 bits which must be set at 4h. Only the top four bits are allowed for edit. These define the channel number

Channel 0: CSR = 14
Channel 1: CSR = 24
Channel 2: CSR = 44
Channel 3: CSR = 84

Frequency Tuning Word

The frequency tuning word (CFTW0) is determined using the following equation:

$$CFTW0 = (\textit{Output Frequency} \times 2^{N})/SYSCLK$$

where: N = 32
SYSCLK = 500 MHz
Output Frequency = required channel frequency MHz

The result must be rounded and converted to hex

e.g. for 51MHz output, CFTW0 = 438086664 = 1A1CAC08 h

Phase Offset Word

Phase Offset Word (CPOW0) is scaled to 14bits , (16383)

CPOW0 = Deg/360 x 16383

The result must be rounded and converted to hex

e.g. 180 deg phase shift on this channel, CPOW0 = 8192 = 2000 h

.

Cont'd

Amplitude Control Register

This ACR register is 24 bits wide but only the lower10 bits (1023) are used for amplitude scaling. Bit-12 must be set high to enable scaling. Other bits are set to 0

Amplitude = A%/100 x 1023

e.g. 70% scaling on this channel

ACR[11:0] = 712 = 2CC h

ACR [23:12] = 001 h

Thus, ACR = 0012CC h

13: TFR and IST tables can be overwritten subsequently without downloading the entire image file.

Example file

This file contains 8 images, each of 256 points and 2 channels per point. The TFR is programmed for External Clock; Inter-Image Delay mode with a 5msec delay programmed after each image. The sequence table is populated with 15 entries running from image 0 to image 7 then in reverse back down to image 0. Each image is played once except for image 7 which is repeated 10 times.

The file is written as a byte stream and stored big-endian. The file identifier at the start of the file, "iHHS_IMG", will confirm if the low-high byte order is correct.

File identifier

```
Address
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   69 48 48 53 5F 49 4D 47 00 01 00 08 00 04 00 00  iHHS_IMG........
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040   00 00 01 00 00 00 00 00 01 00 01 00 00 00 20 10  .............. .
00000050   02 00 01 00 00 00 40 20 03 00 01 00 00 00 60 30  ......@ ......`0
00000060   04 00 01 00 00 00 80 40 05 00 01 00 00 00 A0 50  ......€@...... P
00000070   06 00 01 00 00 00 C0 60 07 00 01 00 00 00 E0 70  ......À`......àp
```

Identifier always starts at address 00000000 h

File Identifier Protocol Version number Number of Images in Bank

Timing Function register (TFR)

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   69 48 48 53 5F 49 4D 47 00 01 00 08 00 04 00 00 iHHS_IMG........
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040   00 00 01 00 00 00 00 00 01 00 01 00 00 00 20 10  .............. .
00000050   02 00 01 00 00 00 40 20 03 00 01 00 00 00 60 30  ......@ ......`0
00000060   04 00 01 00 00 00 80 40 05 00 01 00 00 00 A0 50  ......€@...... P
00000070   06 00 01 00 00 00 C0 60 07 00 01 00 00 00 E0 70  ......À`......àp
```

0004 = EXT_CLK with IMG_DLY

Image Index Field

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   69 48 48 53 5F 49 4D 47 01 00 08 00 00 00 00 00  iHHS_IMG........
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040   00 00 01 00 00 00 00 00 01 00 01 00 00 00 20 10  .............. .
00000050   02 00 01 00 00 00 40 20 03 00 01 00 00 00 60 30  ......@ ......`0
00000060   04 00 01 00 00 00 80 40 05 00 01 00 00 00 A0 50  ......€@...... P
00000070   06 00 01 00 00 00 C0 60 07 00 01 00 00 00 E0 70  ......À`......àp
00000080
```

Index field always starts at address 00000040 h

Index Length in Points Memory Offset Address

For clarity, only indices for images 0, 1 and 7 are highlighted.
In this example each image has the same total byte length of 8208 bytes (2010 h)
i.e. (256 points x 2 channels + 1 header) x 16

Note: Memory Offset Address = [File location – 1000] h

Image Sequence Table (IST)

Follow immediately after the Image Index field (in this example starting at address 00000080 h)

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  69 48 48 53 5F 49 4D 47 01 00 08 00 00 00 00 00  iHHS_IMG........
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040  00 00 01 00 00 00 00 00 01 00 01 00 00 00 20 10  .............. .
00000050  02 00 01 00 00 00 40 20 03 00 01 00 00 00 60 30  ......@ ......`0
00000060  04 00 01 00 00 00 80 40 05 00 01 00 00 00 A0 50  .......€@...... P
00000070  06 00 01 00 00 00 C0 60 07 00 01 00 00 00 E0 70  ......À`......àp
00000080  00 01 00 14 00 1E 00 32 01 01 00 14 00 1E 00 32  .........2......
00000090  02 01 00 14 00 1E 00 32 03 01 00 14 00 1E 00 32  .2.......2......
000000A0  04 01 00 14 00 1E 00 32 05 01 00 14 00 1E 00 32  .2.......2......
000000B0  06 01 00 14 00 1E 00 32 07 0A 00 14 00 1E 00 32  .2.......2......
000000C0  06 01 00 14 00 1E 00 32 05 01 00 14 00 1E 00 32  .2.......2......
000000D0  04 01 00 14 00 1E 00 32 03 01 00 14 00 1E 00 32  .2.......2......
000000E0  02 01 00 14 00 1E 00 32 01 01 00 14 00 1E 00 32  .2.......2......
000000F0  00 01 00 14 00 1E 00 32 FF 00 00 00 00 00 00 32  .2.......2ÿ.....
00000100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

INDEX REPEAT SCR_DLY INT_OSC IMG_DLY

The IST (Sequence table) above lists 15 entries running from image 0 to image 7 then in reverse back down to image 0. Each image is repeated once except for image 7 which is repeated 10 (0A h) times. The sequence stops on completion
(The first, last and middle image sequences are highlighted)

All other parameters are the same for each entry

SCR_DLY = 0014 h = 20 dec x 0.1usec = 2usec

INT_OSC = 001E h = 30 dec x 1usec = 30usec period

IMG_DLY = 0032 h = 50 dec x 0.1msec = 5msec

FF 00 = STOP at end of sequence table (an “FF” command must be included)

Image Data
Always starts at file address 00001000 h

In this example all 8 images are the same and the data for each point in the image is also repeated.

Each image = 256 points comprising 2 channels at 16 bytes per channel
Each image starts with one 16 byte header

In this example total bytes per image = 256x2x16+16 = 8208 bytes (2010 h)

File start address = 4096 (00001000 h)
File end address = 4096+65664 = 69670 (0001107F h)

Image 0, Header
Image 0, Point 1, Channel 0
Image 0, Point 1, Channel 1
Image 0, Point 2, Channel 0
Image 0, Point 2, Channel 1

```
Address
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000FF0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00001000   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
00001010   4F FF 00 12 04 43 80 86 64 05 81 92 06 51 19 19  Oÿ...C€†d..’.Q..
00001020   30 00 00 22 04 57 55 25 17 05 40 96 06 46 07 07  0.."·WU%..@–.F..
00001030   4F FF 00 12 04 43 80 86 64 05 81 92 06 51 19 19  Oÿ...C€†d..’.Q..
00001040   30 00 00 22 04 57 55 25 17 05 40 96 06 46 07 07  0.."·WU%..@–.F..
```

🡻 (further 254 points of Image 0)

```
00003010   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
```

🡻 (256 points of Image 1)

```
00005020   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
```

🡻 (Image 2)

```
00007030   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
```

🡻 (Image 3)

```
00009040   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
```

🡻 (Image 4)

```
0000B050   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
```

🡻 (Image 5)

```
0000D060   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
```

🡻 (Image 6)

```
0000F070   42 5F FF 00 F4 01 D0 00 00 02 20 00 03 00 30 00  B_ÿ.ô.Ð... ...0.
```

🡻 (initial 254 points of Image 7)

```
00011040   4F FF 00 12 04 43 80 86 64 05 81 92 06 51 19 19  Oÿ...C€†d..’.Q..
00011050   30 00 00 22 04 57 55 25 17 05 40 96 06 46 07 07  0.."·WU%..@–.F..
00011060   4F FF 00 12 04 43 80 86 64 05 81 92 06 51 19 19  Oÿ...C€†d..’.Q..
00011070   30 00 00 22 04 57 55 25 17 05 40 96 06 46 07 07  0.."·WU%..@–.F..
```

(end)