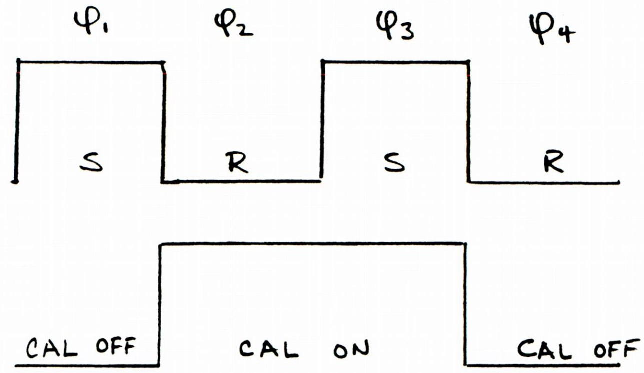
PRJ
4/27/86

Digital Backend Signal Processing

Switch Phases:



$$\text{RMS} = \sqrt{\frac{(\bar{Z} - Z_i)^2}{N(N-1)}}$$

One observing cycle consists of 4 phases

φ_1	signal	$= S$	φ_3
φ_2	ref. + cal	$= R + C$	
φ_3	sig. + cal	$= S + C$	φ_1
φ_4	ref.	$= R$	

SWITCHED POWER SP $[\varphi_1 - \varphi_2] + [\varphi_3 - \varphi_4]$

$= [S - R - C] + [S + C - R] = 2[S - R]$

TOTAL POWER TP $= \varphi_1 + \varphi_2 + \varphi_3 + \varphi_4 = 2[S+R] + 2C$

$RC = \varphi_2 - \varphi_4$ $SC = \varphi_3 - \varphi_1$

CAL VALUE CV $= \varphi_2 - \varphi_4 + \varphi_3 - \varphi_1 = 2C$

$CV = SC + RC$

ZERO LEVEL ZL $= [\varphi_1 - \varphi_2] + [\varphi_4 - \varphi_3]$

$= [S - R + C] + [R - S - C] =$ "0"

Data samples will consist of an integration of many phase cycles. With the present system, only the integrated samples will be stored, i.e., individual phases and cycles will not be saved. ~~Considering the case of an ON/OFF, data storage could take two forms~~

The most general data storage method is to save each phase in raw form and to perform the computation of SP, TP, CV, ZL in analysis. If φ_{21} denotes "phase 2, sample 1", then the data could be stored either as

$$\varphi_{11}, \varphi_{21}, \varphi_{31}, \varphi_{41}, \varphi_{12}, \varphi_{22}, \varphi_{32}, \varphi_{42}, \cdots$$

or

$$\varphi_{11}, \varphi_{12}, \varphi_{13}, \cdots \varphi_{21}, \varphi_{22}, \varphi_{23}, \varphi_{24}, \cdots, \varphi_{31}, \varphi_{32}, \cdots$$

The first option is the best, although if an observer dumps the data array, he may have trouble making sense of it.

Switched Power Data Calibration

The cal value can be computed in several ways

(1) Each cal sample applied individually to its associated switched power sample

(2) All cal samples averaged and the same average sample applied to each SP sample

(3) A least squares fit performed to the array of cal sample against sample number and the LSQ values applie

The conservative approach is to take the average of the cal samples. The switched power antenna temp for sample i becomes

$$T_i^{SP} = \frac{SP_i}{\langle CV \rangle} TC = \frac{2[S-R]_i}{\langle 2C_i \rangle} TC$$

where TC is the temperature of the cal signal

A POPS procedure can be written to perform this calculation. The total amount of data stored per ON/OFF scan will be

$$(4 \text{ phases}) \times (4 \text{ samples/repeat}) \times (\# \text{ repeats})$$

$$= 16 \times RPT \qquad (RPT \leq 6)$$

PRJ
4/30/86

1 – 8   9 – 16   17 – 24   25 – 32

1 – 12   13 – 24   25 – 36   37 – 48

SP₁ = TWH(129+I, 2)

PH1 = TWH(128+I, ~~RTWH~~ 2)
PH2 = TWH(129+I, 2)
PH3 = TWH(130+I, 2)
PH4 = TWH(131+I, 2)

PROCEDURE DIGBAK4

TRT

RPT = 1

```
NS = NUMBER OF SAMPLES
RPT = NS/4.
FOR I = 1 TO NS
  J = 128 + I
  PH1 = TWH( J , 2)
  PH2 = TWH(J+1, 2)
  PH3 = TWH(J+2, 2)
  PH4 = TWH(J+3, 2)
  TWH( J , 1) = PH1 - PH2 + PH3 - PH4
  K = J + RPT * 4.          133
  TWH(K, 1) = PH1 + PH2 + PH3 + PH4
  L = J + RPT * 8.          137
  TWH (L, 1) = PH2 - PH4 + PH3 - PH1
  M = J + RPT * 12.
  TWH (M, 1) = PH1 - PH2 + PH4 - PH3
END
RETURN
FINISH
```

SP 129 130 131 132

TP 133 134 5 6

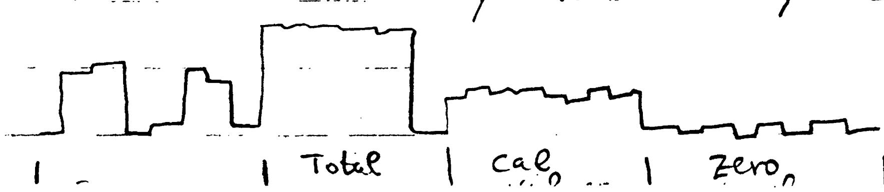
CV 137 8 9 140

ZL 141 142 143 144

141
129
12
141

TEMP ARRAY CONTAINS ALL SWITCHED POWER SAMPLES followed by TOTAL POWER SAMPLES, CAL SAMPLES, ZERO SAMPLE

Data array



(ELEMENT)

```
PROCEDURE SP (XSCAN)
GET (XSCAN)
DIGBAK4

SLABEL=1; PAGE SHOW SLABEL=0; AVG
FINISH


PROCEDURE
```

Procedure DIGBAK4 takes a 4-phase digital backend scan and computes switched power, total power, cal, and zero level signals and stores them in a single scan. (sw. pwr signals consecutive, etc.)

Define a procedure SP that calls DIGBAK4 and displays the switched power signal. Similarly, let TP display the total power signal, CV the cal value, and ZL the zero level

After any one of procedures SP, TP, CV, or ZL has been called, all the data are present, so there is no need to fetch the raw data from the disk to compute one of the other quantities. Define other procedures SPX, TPX, CVX, ZLX that display the appropriate quantities from the currently referenced array.