

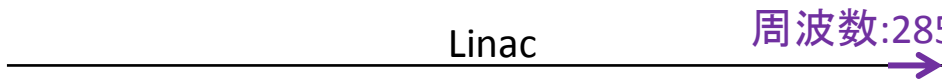
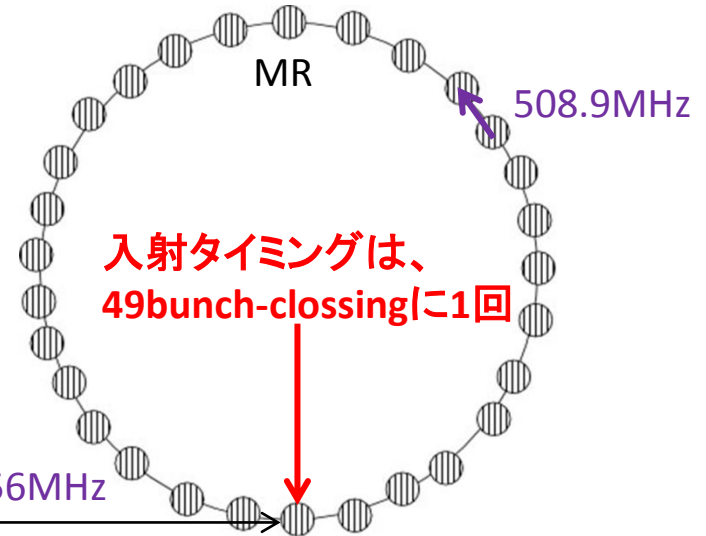
SuperKEKB用Bucket selection案

梶 裕志

Bucket selection at KEKB

Linac timingにdelayを与えることで、
入射するMR-bucketを選択

しかしLinac-RF(周波数:2856MHz)とMR-RF(508.9MHz)
の位相が揃う頻度は10.38MHz
これが入射タイミングの頻度で、MR-RF周波数の1/49

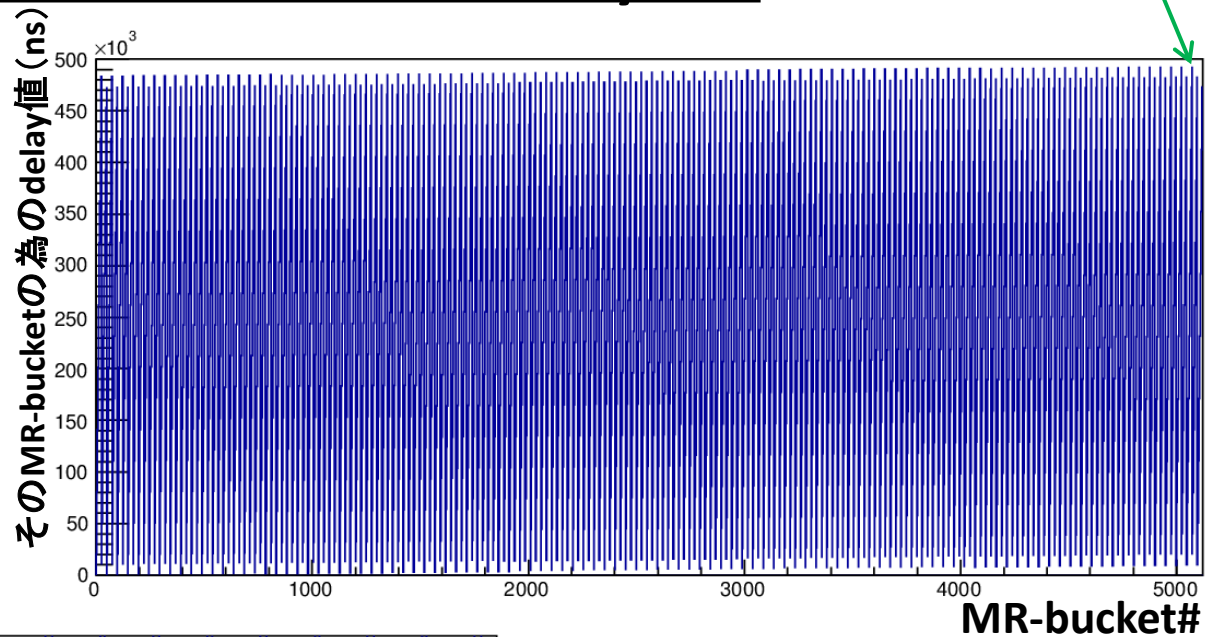


入射機会(回目)	Delay値	MR-bucket#
1	0 ns	0
2	96 ns	49
3	192 ns	98
..
105	10.0 μs	25
..
209	20.0 μs	1
..
5120	493μs	5071

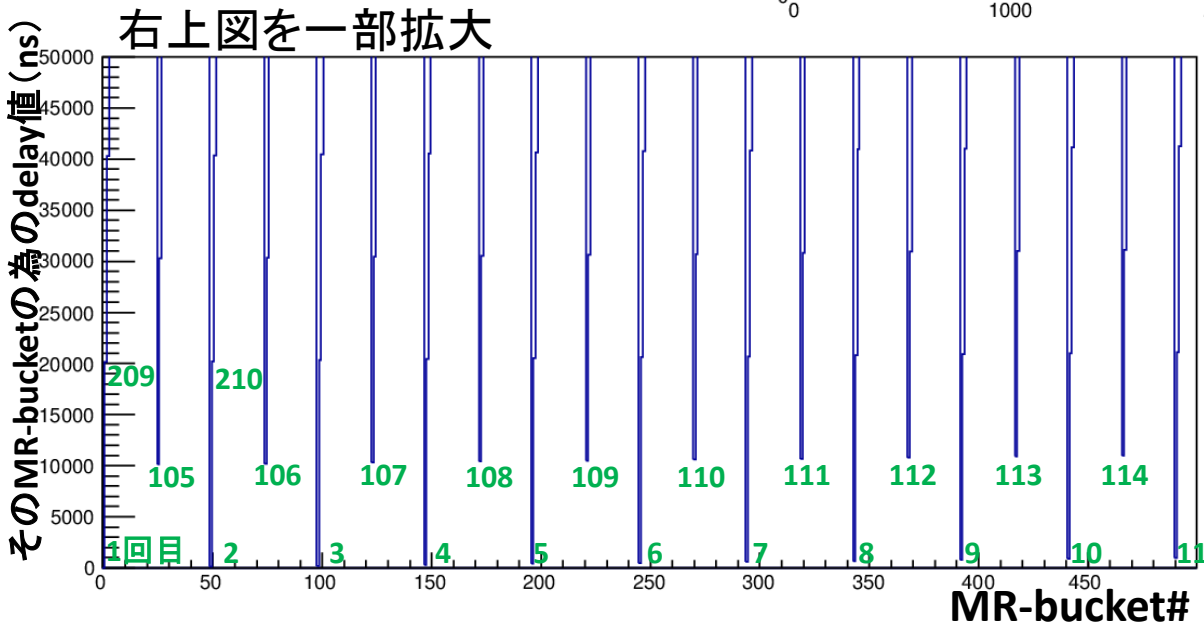
Delay=0で入射できる
MR-bucketを#0と定義

MR-bucket# vs delay値

5120回目に
#5071を選択



KEKB(電子・陽電子)で
MR各bucketの為のdelay値



<500 μ sで全bucketを選択可能

SuperKEKBでも
電子入射はこれに従う

SuperKEKB陽電子の場合

陽電子はDR (RF周波数:508.9MHz)を通して入射
 ⇒ MR-bucketごとに
 「前述のdelay値で出射できるDR-bucket」を使う。



入射機会(回目)	Delay値	MR-bucket#	DR-bucket#
1	0 ns	0	0
2	96 ns	49	49
3	192 ns	98	98
4	288 ns	147	147
5	384 ns	196	196
6	480 ns	245	15
..
5120	493 μ s	5071	131

Delay=0で出射できる
DR-bucketを#0と定義

これらのDR-bucketを選べば、
KEKBと同じdelay値で入射可能

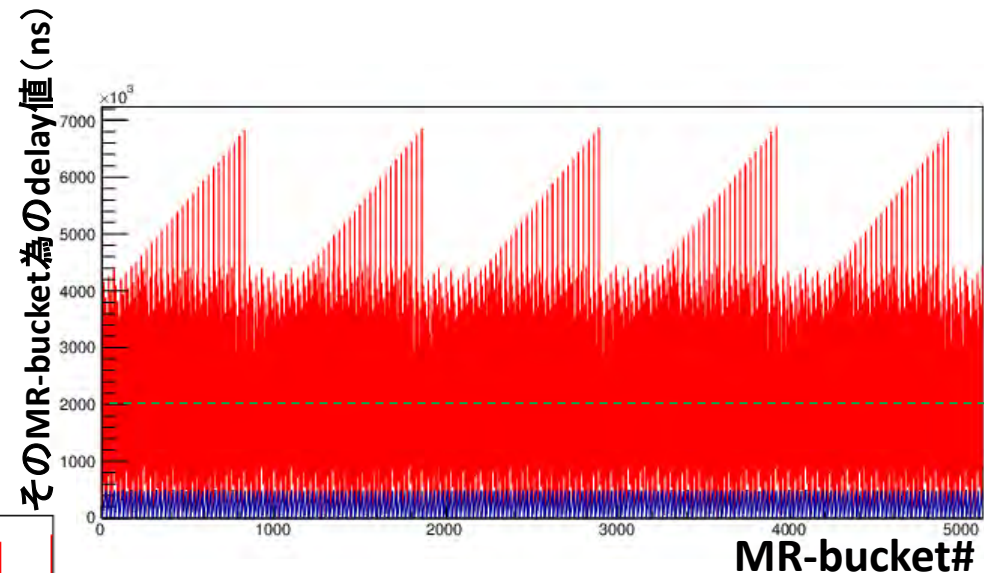
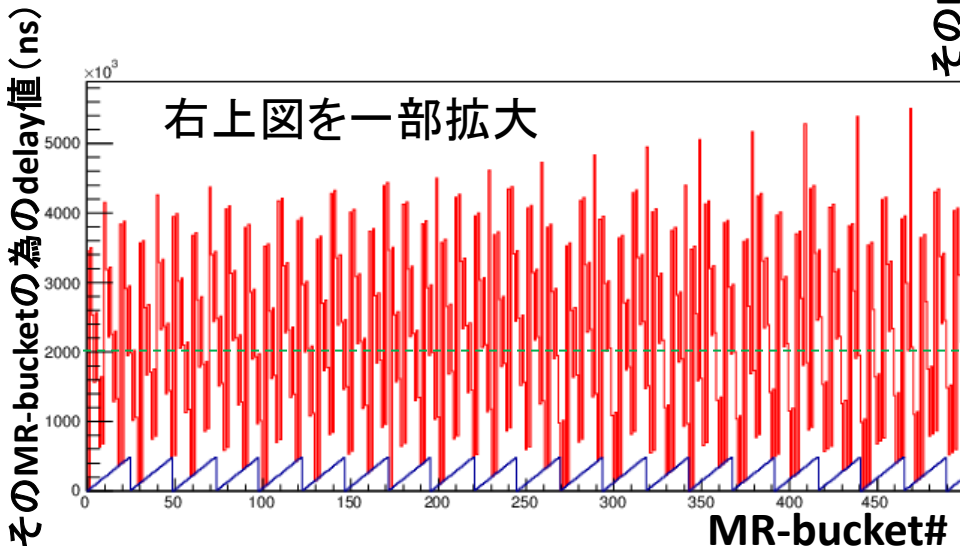
陽電子入射でDR-bucket選択に課される制限

現実には入射したいMR-bucket#に都合の良いDR-bucket#が選べないこともある。

例えばDR-bucket# = 0, 49が埋まっているとき

次の入射のために使えるDR-bucketは#99-#180(1-bunch), #99-#129(2-bunch)

DR-bucketを#99-#129しか選べないとき、
各MR-bucketを選択するためのdelay値
(青はP3と一緒に)



調度良く使えるDR-bucketが埋まっているとき、
次のtimingまで待たないといけないため、
delay値が大きくなる。

特にSuperKEKBで可能な**最大delay値(2ms)**以上
を必要とするbucketが2323bucketsある。

現在考えられる対処法

Bucket selection内のdelay計算routineは以下のオプションに対応できる様にしておく。

Linac前半を空撃ち

DR \Rightarrow MR入射だけを行い、DRを空にする

- T=0直後に必ず必要なオプション
- その分dead timeが生まれる

Linac後半のRF位相を変調(508.9MHz clock単位)

- 例えば「DR:#1 \Rightarrow MR:#1」通常入射では \Rightarrow 'delay=3.5ms'
- RF位相を+2ns(+1clock)すれば \Rightarrow 'delay=2ns'
- T=0直後は無理?

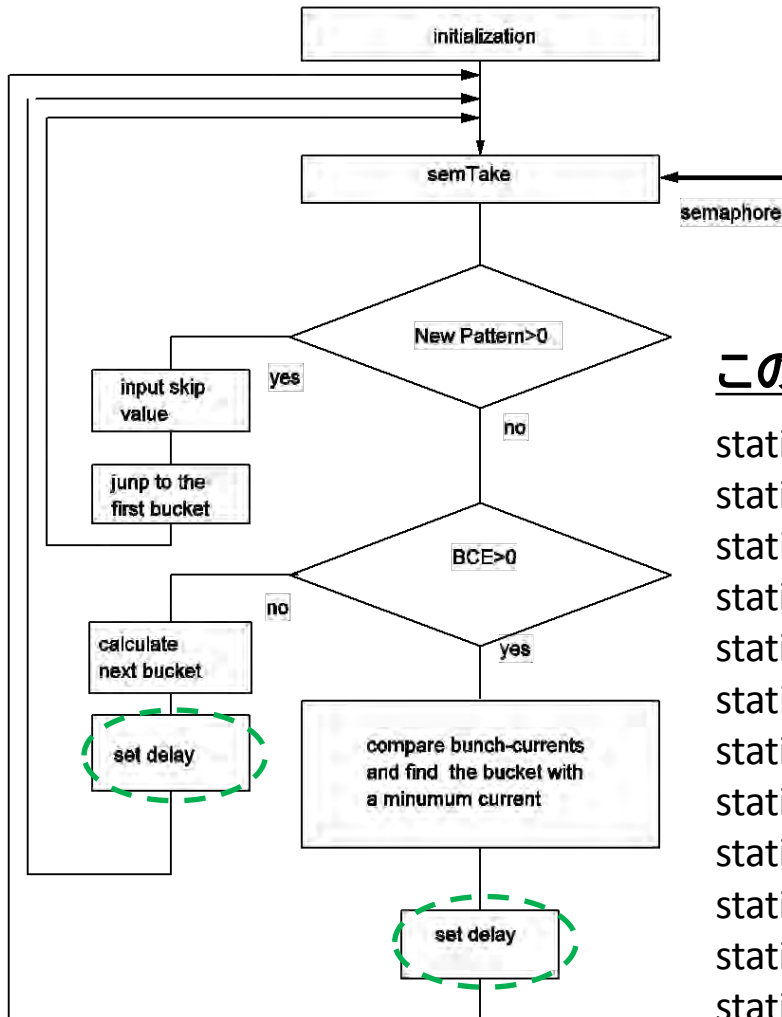
DRで先出し後入れ

- まず始めの500 μ sで「DR \Rightarrow MR」、DRを空にする
- 「Linac前半 \Rightarrow DR」は必ずdelay>500 μ sとし、500 μ s以内に「DR \Rightarrow MR」できるDR-bucketを選ぶ
- Linac前半でdelay>500 μ sが必要
- T=0直後は無理?

Bucket selection全体像

今日のところは「set delay」部分だけを議論

- 入射bucketが決定した後にそのdelay値を計算し、各機器にセットする。
- 以下の仕様を満たすものにする。
「1-bunch/2-bunchどちらでも対応」
「前頁の3対処法すべてに対応」



このプログラムでのglobal (public) 変数 - 案 -

static int eleposi;	=0(電子), =1(陽電子)
static int nobunch;	1-bunch/2-bunch入射
static int dr_mask[230];	=0(入射できないDR-bucket)
static int delay_max;	delayの最大値
static int n_fill;	次に入射するMR-bucket#
static int delay0;	Linac前半のdelay値を一時保管
static int delay1[5];	Linac後半のdelay値を一時保管
static int dr_cand[5];	そのとき使うDR-bucket#
static int ngood;	条件を満たすdelay候補#
static int delay_gun;	次の20msのLinac前半delay値
static int delay_dr;	次の20msのLinac後半delay値
static int delay_dr1;	次の次の20msのLinac後半delay値
static int offset_rf;	次の20msのLinac後半RFの位相変調値

Example of delay calculation

```

/* Following two work as input variables */
static int dr_mask[230];
static int delay_max;
/* Results are stored into following three variables */
static int delay1[5];
static int dr_cand[5];
static int ngood;
int calcu_delay(int bucket_n, int n_offset){

    int i,j;
    int n_tot;
    int n_inj;

    if(n_offset<0) return(-1);
    if(bucket_n<0 || bucket_n>5119) return(-1);

    i=0;
    j=0;
    while( i<5 ){

        n_tot = j*49+n_offset;
        n_inj = n_tot%5120;
        if(n_inj == bucket_n){
            delay1[i] = n_tot;
            dr_cand[i] = n_tot%230;
            i++;
        }
        j++;
    }

    ngood=0;
    for(i=0; i<5; i++){
        if(dr_mask[dr_cand[i]]){
            if(delay1[i]<delay_max){
                ngood = i;
                return(1);
            }
        }
    }
    return(0);
}

```

MR-bucketを選ぶためのdelay値(508.9MHz単位)を5つ計算しglobal変数“delay1[5]”に保存

- 以下2つがinput値
 - bucket_n: 選ぶMR-bucket
 - n_offset: Linac後半RF位相のoffset (508.9MHz clock単位)
- その候補が使うBR-bucketも“dr_cand[5]”に保存

さらに候補が以下の条件を満たすか調べる

- delay値が条件“delay_max”以下
- BR-bucketが入射可能“dr_mask[i]==1”
- 条件を満たす候補が見つかった場合その番号をglobal変数“ngood”に保存

routine自身は以下の値を返す

- 条件を満たす候補が見つかった場合: 1
- 条件を満たす候補が見つからなかった場合: 0
- Input値がおかしい場合: -1

Example of delay calculation for DR

```
/* Result is stored into the following variable */
static int delay0;
int calcu_delay_dr(int bucket_dr){

    int i;
    int n_tot;
    int n_inj;

    if(bucket_dr<0 || bucket_dr>229) return(-1);

    i=0;
    while( i<230 ){

        n_tot = i*49; ←ここにoffsetが足される
        n_inj = n_tot%230;
        if(n_inj == bucket_dr){
            delay0 = n_tot;
            return(1);
        }
        i++;
    }

    return(0);
}
```

- DR-bucketを選ぶためのdelay値(508.9MHz単位)を計算しglobal変数“delay0”に保存
- 入射したいDR-bucket#がinput値
 - delayなしで出射できるbucketを#0としたので、本当はこの値にoffsetがあるはず。

Linac前半を空撃ち

```

static int n_fill;    /* MR-bucket to be filled */
static int delay0;
static int delay1[5];
static int dr_cand[5];
static int ngood;
void main_normal(void) {

    int flg1 = 0;
    int flg2 = 0;

    while( 1 ){

        semTake(syncSem, WAITFOREVER);

        flg1 = calcu_delay(n_fill,0);
        if(flg1==1){
            flg2 = calcu_delay_dr(dr_cand[ngood]);
        }

        if(flg1==1 && flg2==1){
            set_delay_p(delay0,delay1[ngood],0);
        }else{
            set_delay_p(-1,-1,0);
        }

    }

    set_delay_pと連動して、
    delay0を基にdr_mask[230]をアップデート
    するprogramも必要

```

ここにはまだないが、本当は、「入射MR-bucket#を決定し、それをn_fillに入れる」部分が必要

Linac後半のdelay値を計算

- RF変調ができないときは(例えばT=0では)、2つ目の引数を常に0とすれば良い。
- 前述の通り、結果はdelay1[5], dr_cand[5]に保存される。
- このprogramは電子bucketのdelay値計算にも使える。(常にdelay1[0]を使えば良い)

Linac前半・後半のdelay値をセット

- このprogramの作成が必要
- 例えば「次のプロセスでLinac前半は動かない」時は引数を負にしておく。

DRで先出し後入れ

```

static int n_fill; /* MR-bucket to be filled */
static int dr_mask[230];
static int delay0;
static int delay1[5];
static int dr_cand[5];
static int ngood;
void main_eoli(void) {

    int flg1, flg2;
    int n_offset0 = 254800; /* correspond to 509600 ns */

    while( 1 ){

        semTake(syncSem, WAITFOREVER);

        flg1 = calcu_delay(n_fill, 0);
        flg2 = calcu_delay_dr(dr_cand[0]);
        if(flg1==1 && ngood==0){
            set_delay_p(delay0, delay1[0], 0);
        }else{
            set_delay_p(delay0+n_offset0, delay1[0], 0);
        }
    }
}

```

必ず最短delay値で取り出せるDR-bucketを選び続ける。

- 毎回、DRを空にしてから次の陽電子を入射
- そのDRが選べないときは、delay値を+509.6 μ sする。
 - このoffsetの値は許されるdelayの最大値を考慮して決める。

Linac後半のRF位相を変調

```
static int n_fill; /* MR-bucket to be filled */
static int delay0;
static int delay1[5];
static int dr_cand[5];
static int ngood;
void main_rfphase(void) {

    int i;
    int flg1 = 0;
    int flg2 = 0;
    int delay_best = 1000000;
    int dr_best;
    int offset_best;

    while( 1 ){

        semTake(syncSem, WAITFOREVER);

        for(i=0; i<49; i++){
            flg1 = calcu_delay(n_fill,i);
            if(flg1==1){
                if(delay1[ngood]<delay_best){
                    delay_best = delay1[ngood];
                    dr_best = dr_cand[ngood];
                    offset_best = i;
                }
            }
        }

        calcu_delay_dr(dr_best);
        if(flg1==1 && flg2==1){
            set_delay_p(delay0,delay1[ngood],offset_best);
        }else{
            set_delay_p(-1,-1,0);
        }
    }
}
```

offsetを変えながらloopし、
その中でdelay値が最短になるものを選ぶ

Delay値の設定 - 案 -

電子・陽電子どちらの場合もdelay_gun, delay_drがLinac前半, Linac後半のdelay値になる。
「revolution/49を受け取り、それを基準にdelay値だけ遅れて動作」ことを仮定

電子

```
static int delay_gun;
static int delay_dr;
static int offset_rf;
void set_delay_e(int clk0){
```

```
    delay_gun = clk0;
    delay_dr = clk0;
    offset_rf = 0;
```

delay_gunをLinac前半に設定
delay_drをLinac後半に設定
(上記2つは値が負のとき動作しない様にする)
offset_rfをLinac後半RF位相の変調値に設定

```
}
```

陽電子

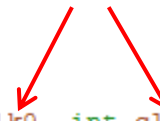
```
static int delay_gun;
static int delay_dr;
static int delay_dr1;
static int offset_rf;
void set_delay_p(int clk0, int clk1, int clk_offset){
```

```
    delay_gun = clk0;
    delay_dr = delay_dr1;
    delay_dr1 = clk1;
    offset_rf = clk_offset;
```

delay_gunをLinac前半に設定
delay_drをLinac後半に設定
(上記2つは値が負のとき動作しない様にする)
offset_rfをLinac後半RF位相の変調値に設定

```
}
```

前頁までで計算したdelay値を
inputにする



部分はまだ未完成だが、関係各所と話し合っ仕様・アルゴリズムを決めたい。

Bucket selection仕様まとめ - 案 -

Delay値計算の方法(P8, P9)は今日の案で確定したい。

- Delay値は5つ計算し、その中から可能なdelay値を選ぶ。
- DR-bucket使用の可否も判定する。
- Linac後半RF位相の変調値もinput値になっている。

基本的に以下の方針で、

「delay値決定(P10, P11, P12)」「その設定(P13)」routineを1つのprogramにまとめる。

- 指示されたMR-bucketに入射できないときは「空撃ち」する。
- 「DRで先出し後入れ」「Linac後半のRF変調」は、if文で場合分け。

「入射bucketを決め、上述programにdelay値を計算」させるmain programを作成

- 「BCE/non-BCE」「1-bunch/2-bunch」もここで決定
- 「DRで先出し後入れ」「Linac後半のRF変調」を指定可能にする。
- 外部からのinput値で、オペレータもそれを変更可能にする。

エキスパートが、入射方法の特徴を熟知し、このprogramをメンテナンスする。

- 陽電子入射を50Hzで行うならば、「DR先出し後入れ」「Linac後半RF変調」の両方が必要
- 「空撃ち」の頻度を抑える様に、programを最適化

Bucket selectionのoutput値 - 案 -

各機器に与えるタイミング情報は以下の仕様にしたい

- すべて508.9MHzのカウント数
- revolution/49信号からのdelay値
- Delay値<0のときは次の20msでは動作しない(陽電子空撃ち用)。

これとは別にLinac後半RFの位相の変調値も与える。

- 508.9MHzのカウント数
- 変調しないときは“0”を渡す ← T=0でもこの値を受け取ってほしい

Summary

KEKBでは、
最大500 μ sの入射delayにより、電子・陽電子の全bucketを選択可能

SuperKEKBでは、

- 電子入射はこれに従う。
- 陽電子入射は(適したDR-bucketが使えないとき) delay値が大きくなる。

陽電子のdelay計算アルゴリズムを考え、sourceの例を作成した。

- delay値は5つ計算し、その中から可能なdelay値を選択
- 「DR先出し後入れ」「Linac後半のRF変調」の場合にdelay値を計算可能
- 可能なdelay値がない場合は、陽電子を空撃ちする。
- Output値は「Linac前半/後半のdelay値」「Linac後半のRF変調値」

今後は以下のプログラムも製作する。

- Bucket selectionのmain program
- Delay値を各機器に受け渡すprogram
- 入射できないDR-bucketをマスクするprogram
- 陽電子の50Hz入射に対応

そのために各機器の担当者に提案し、仕様を決定したい。