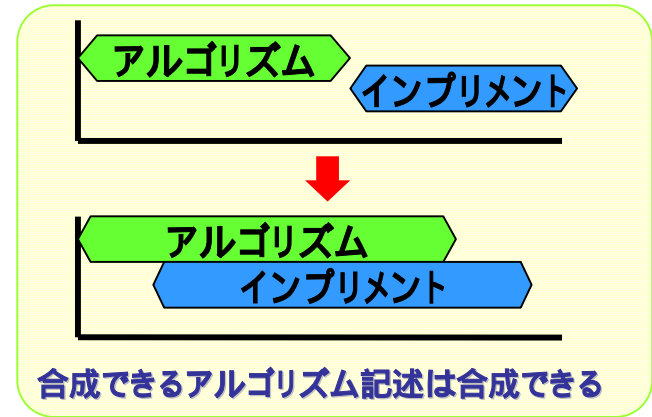
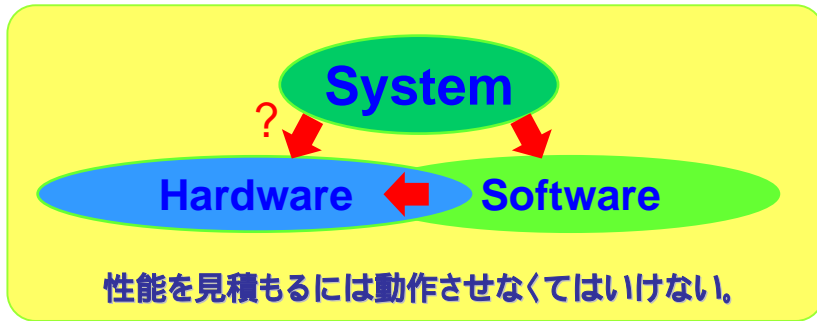


Cベース設計を上手に使いこなそう！



```

struct bar {
  int addr;
  bar(int addr_i):addr(addr_i){}
  int operator=(const bar &other){return(*this=(int)other);}
  int operator=(int data){ return(BUS_WRITE(addr,data));}
  operator int() const { return(BUS_RD(addr)); }
  int operator--(int) { int tmp=(int)*this;
                      *this=tmp-1; return(tmp);}
  ...
};
struct foo {
  int base;
  foo(int base_i):base(base_i){}
  ...
  bar operator[](int addr) const {
    return bar(addr+base); }
};
    
```

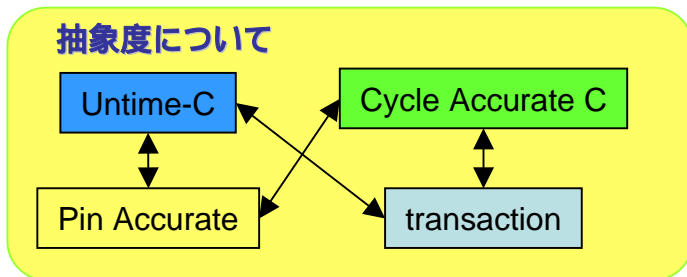
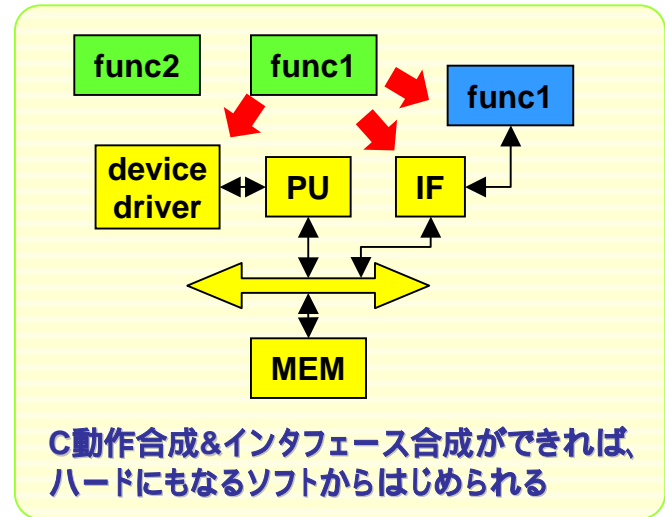
func2

func1

?

MEM

ポインタ、参照演算子のオーバーロードを利用してシステムシミュレーションのための記述変更を減らす。



計算量を把握して高速化

プロセッサ速度

シミュレーション速度 = $\frac{\text{プロセッサ速度}}{\text{モデルの計算量}}$