CP-PACS プロジェクトに携わって

山下　義行
筑波大学電子・情報工学系（当時）
佐賀大学理工学部知能情報システム学科（現在）

プロジェクトから離れてもうだいぶ経ちますが，未だに様々な事柄を鮮明に思い出します．先日，当時指導していた大学院生と久しぶりに食事をする機会があり，当時のいろんな思い出話に盛り上がりました．話していて分かったことは，CP-PACSの思い出というと二人共，同じ話に収束してしまい，それというのは，スライド・レジスタを導入するとループ最適化法がこんなにも劇的に変化するものなのかと驚いたことでした．さらに私について言えば，未だにそれを引きずっています．以下，とても限定された話題ですが，記録に残していただければ幸いです．

まず，理論的に言えば，命令スケジューリングもレジスタ割り付けも一般にNP問題であって複雑なアルゴリズムを用いてなお準最適解しか求めることができない厄介な問題です．しかし，スライド・レジスタを用いるならば，いずれもＰ問題になるため，ほぼストレートに最適解に到達できます．特にレジスタ割り付けについては，当初，NPと思い込んでいたためにＰと知って驚愕したことを憶えています．この結果はさっそく1本の論文にまとめたのですが，論文の書き方が悪かったせいか（当時，諸事情で落ち着いて論文を書く余裕がありませんでした．尤も今も落ち着いて論文を暇はありませんが），査読者から「そんなことはあり得ない」と一喝されて不採録になってしまいました．しかしそれすら「査読者がそう断定しても不思議ではない」と妙に納得したものでした．スライド・レジスタの理屈はとても簡単で，スライド・レジスタを用いればループ内でのレジスタの自己干渉を自然に解消でき，組み合わせ問題の最適化に付きまとう様々な煩わしさが跡形も無く消えてしまいます．結果，プログラムのループ部を基本ブロック部であるかのように取り扱うことができるのですが，その辺りに不思議な感覚が残ります．

実用上も，スライド・レジスタを用いることで命令スケジューリングの最適

化アルゴリズムが劇的に単純化されます．これには助かりました．当時のプロジェクト内での私の役割は，最適化アルゴリズムをコンパイラに試験的に実装してみせることでした．本来 NP であるような問題に取り組むとなると，検討段階では単純なアルゴリズムでおおむね問題ないと思っていても，実装段階では予想外の経験値を導入せざるを得なくなったりして次第に複雑怪奇なものに変容していくものです．しかしスライド・レジスタを使ってプログラミングしてみると「これでいいのだろうか」と思うくらいに単純な実装でも十分であることが分かり，調子抜けしたことを印象深く憶えています．短期間で最適化プログラムを試作できたことはスライド・レジスタあったればこそだったでしょう．

さて，未だにコード最適化の研究をしています．ちょうどプロジェクトを離れる頃に，Intel の Itanium プロセッサが rotating register という機構を持つことを知りました．これは CP-PACS のスライド・レジスタとほとんど同じものです．今ではこのプロセッサの上でいろんなプログラムを組んではぽつぽつと論文を書いています．前出の大学院生とは不採録の論文を書き直そうと約束していますが，果たしていつのことになるのか．残念なことに Itanium プロセッサは商業的には失敗し，市場から消えそうです．消えてしまう前にはすっきりとスライド・レジスタ関連の研究に全て決着をつけたいと考えていますが．