مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (4) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (4) 2016

تطوير تقنية هجينة للإدارة الديناميكية للطاقة في المعالجات متعددة النوى

الدكتور محمد حجازية*
يوسف نتيفة**

(تاريخ الإيداع 4 / 7 / 2016. قُبِل للنشر في 23/ 8 / 2016)

□ ملخّص □

أدى تطور التكنولوجيا إلى تحسن كبير في أداء الحاسبات و الأجهزة المحمولة و ترافق ذلك مع ازدياد في استهلاك الطاقة مما جعل الاهتمام بإدارة استهلاك الطاقة أمر ضروري. يعد المعالج من ضمن العناصر الأكثر استهلاكا للطاقة لذلك يهدف البحث إلى تطوير تقنية جديدة لإدارة الطاقة في المعالجات متعددة النوى التي تدعم مختلف الأجهزة الإلكترونية حاليا، مع الإشارة إلى أن إدارة الطاقة تعد من الأمور الهامة في إطار الأبحاث العلمية المتعلقة بالمعالجات متعددة النوى لأنها تحقق التوازن بين متطلبات الاداء العالي للمعالج و تأثير الاستهلاك المتزايد للطاقة فيه والتأثيرات الحرارية المرافقة و أثرها على موثوقية النظام. تطرق البحث إلى عدة تقنيات مثل (التدرج الديناميكي للجهد والتردد (DVFS)، النوى غير المتناظرة، حركة سلاسل التعليمات ، النوى متغيرة الحجم، دمج النوى) وتم وضع جدول مقارنة بين هذه التقنيات يبين الميزات السلبية و الإيجابية لكل منها و بنتيجة البحث تم اقتراح تقنية هجينة تستفيد من أهم خصائص التقنيات المدروسة دون أن تتعارض فيما بينها.

الكلمات المفتاحية: معالجات متعددة النوى ، إدارة الطاقة ، توفير الطاقة ، تقنية DVFS ، تعدد النوى غير المتناظرة .

* أستاذ مساعد، قسم هندسة الحاسبات والتحكم الآلي، كلية الهندسة الميكانيكية والكهربائية، جامعة تشرين، اللاذقية، سورية.
** طالب دراسات عليا (ماجستير)– قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية و الكهربائية– جامعة تشرين– اللاذقية–سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (4) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (4) 2016

Developing a Hybrid Technique for Dynamic Power Management in multicore processors

Dr. Mohammed Hijazieh*
Youssef Ntefeh**

(Received 4 / 7 / 2016. Accepted 23 / 8 / 2016)

□ ABSTRACT □

Recent technological advances have greatly improved the performance and features of computers and mobile systems. This improvements leads to increase in power consumption which makes the task of managing their power consumption necessary. The processor considered as one of the most power consuming elements in the system so, this research aims to develop a new method for power management in multicore architecture which support most of the modern electronics. Power management techniques is an important field in multicore studies because it must balance between the demanding needs for higher performance/throughput and the impact of aggressive power consumption and negative thermal effects. Many techniques have been proposed in this research like (Dynamic Voltage and Frequency Scaling (DVFS), Asymmetric cores, Thread motion, variable size cores, core fusion) then after we summarized comparing table which clarifies the pros and cons of these techniques, we proposed a new technique for power management in multi-core processors implements the best of these techniques .

Key Words: Multi-core processing, power management, DVFS technique, asymmetric multi-core.

*Assistant Professor, Department of Computers and automatic control, Faculty of mechanical and electrical engineering, Tishreen University, Lattakia, Syria.
***Postgraduate student (Master degree), Department Of Computer And Automatic Control Engineering, Faculty Of Mechanical And Electrical Engineering, Tishreen University, Lattakia, Syria.

مقدمة :

إن موضوع استهلاك الطاقة هو أمر في غاية الأهمية عالميا لتقليل استهلاك الطاقة بالإضافة إلى توفير تكاليف التشغيل. حسب إحصائيات [1] يتسارع اقتناء الحاسبات بشكل كبير حيث استغرق العالم 27 سنة ليشغل المليار الأول من الحاسبات، بينما استغرق سبع سنين لتشغيل المليار الثاني، أي يوجد حول العالم اليوم ملياري حاسب و ملياري هاتف ذكي وفقا ل [1]، و استنادا لتلك الإحصائيات فإن توفير استهلاك الطاقة في هذه المعالجات سيحدث فرقا ذو تأثير كبير. بالإضافة إلى إن زمن تشغيل الأجهزة المحمولة محدد بعدة ساعات وفقا للمدخرة المستخدمة. و يتسارع استهلاك هذه المدخرة بتزايد تعقيد التطبيقات و الحاجة لأداء حاسوبي عال. لذا تم تقديم معالجات قوية متعددة النوى و متغيرة التردد صغيرة الحجم مما أدى إلى إعادة تصميم الأنظمة بحيث تتناسب مع بيئة العمل و تكييف استهلاك الطاقة مع الحاجة الحقيقة الآنية للأداء. و لتجنب الخسارة الكبيرة في الطاقة عند امتلاك اجهزة حاسوبية ذات قدرات عالية يجب تصميمها بالشكل الأمثل الموفر للطاقة. على سبيل المثال يجب أن يكون المعالج قادرا على تخفيض الطاقة فوريا في حالات الخمول و عدم انجاز عمل.

أدى ظهور المعالجات متعددة النوى إلى ثورة في المجالات البحثية وفتح المجال أمام دراسات جديدة. قبل ظهور المعالجات متعددة النوى كانت سرعة المعالجات تزداد بشكل أسي و كلما زاد عدد الترانزستورات كلما زادت السرعة و ارتفع الأداء. و لاحظ العالم مور أن عدد الترانزستورات يتضاعف تقريبا كل سنتين[2]. و مع استمرار زيادة سرعة المعالج و الزيادة الكبيرة في أعداد الترانزستورات التي تبدل حالتها بتردد عال جدا و تستهلك طاقة كبيرة ، لذلك ظهرت الحاجة للمعالجة المتوازية و لم تكن لغات البرمجة و التعليمات و التطبيقات حينها تدعم التنفيذ المتوازي للمهام بشكل كافي، لذا ظهر مفهوم تعدد النوى عام 1996 الذي نص على زيادة عدد النوى البسيطة في رقاقة واحدة عوضا عن تصميم نواة ضخمة معقدة. ساعد هذا التصميم على توزيع تنفيذ البرامج على نوى المعالجة حيث يتم فحص اعتماديتها و قابليتها للعمل على التوازي اولا ثم تسند وفقا لأولوياتها إلى النوى الأكثر تفرغا. تصمم حاليا جميع المعالجات الحديثة ببنية متعددة النوى مما فتح المجال أمام أبحاث مثل: إدارة الطاقة ، تصميم شبكة الاتصال الداخلية بين النوى و مشاكل الموثوقية و تقنيات البرمجة المتوازية. و كان الهدف الأساسي من إدارة الطاقة هو رفع الأداء مع الحفاظ على مستوى الاستهلاك. تم في هذا البحث مناقشة عدد من التقنيات المعروفة و بيان كيفية عملها لتقليل الخسارة في الأداء في حال تخفيض استهلاك الطاقة.

مشكلة البحث:

أثار تصميم معالج متعدد النوى عدد من الأبحاث المتعلقة بهذا المجال، منها كيفية الاستفادة من مفهوم تعدد النوى بتقسيم البرنامج إلى سلاسل تعليمات Threads يمكن تنفيذها بشكل مستقل عن بعضها. و لكن لا يمكن توزيع كل البرامج بهذه الطريقة. حيث أن السرعة القصوى للتنفيذ محدودة بالجزء التسلسلي و يطلق عليها اسم عنق الزجاجة التسلسلي (serial bottle neck)حيث يخفض البرنامج المتسلسل الكفاءة المتوقعة من المعالج و يسبب ضياع للطاقة. كما ان الجزء المتوازي من البرنامج ليس متوازيا تماما لعدة أسباب مثل المزامنة وعدم توازي الحمل وتنافس النوى على المصادر المشتركة[3]. يعد تصميم البنية المتعددة النوى غير المتناظرة من اهم حلول مشكلة توزيع الحمل و تتألف هذه البنية من نوى كبيرة و نوى صغيرة متطابقة بالبنية العامة و بطريقة تنفيذ التعليمات و لكن تختلف عن بعضها (غير متناظرة) بالبنية الدقيقة كترتيب تنفيذ التعليمات و طريقة جدولة المهام و عدد مراحل pipeline، حيث يمكن تسريع الجزء التسلسلي من البرنامج بتنفيذه بواسطة النوى الكبيرة و تنفذ الاجزاء المتوازية من البرنامج بواسطة

النوى الصغيرة. كما يمكن زيادة انتاجية النظام بمشاركة النوى الكبيرة للنوى الصغيرة في التنفيذ في حال عدم وجود جزء تسلسلي. شرح في [6] طرق تطبيق عدم التناظر و تقسم إلى طرق ديناميكية و طرق ساكنة. في الطرق الساكنة يمكن تصميم النوى للعمل بترددات مختلفة أو تصمم نواة أكثر تعقيدا مختلفة بنيويا تماما عن الأخرى. أما في الطرق الديناميكية، يمكن رفع التردد ديناميكيا عند الطلب أو تدمج نوى صغيرة لتشكيل نواة كبيرة [7].

أهمية البحث و أهدافه:

تعد إدارة استهلاك طاقة المعالجات من الأبحاث ذات الأولوية في تطوير المعالجات و العمل الحاسوبي بشكل عام إذ لم تعد القدرة الحاسوبية هي المشكلة فالأجهزة المحمولة و الهواتف تحتوي معالجات ذات قدرات عالية ولكن تتركز المشكلة في كيفية زيادة زمن استخدام هذه الأجهزة دون الحاجة لإعادة الشحن و تقليل استهلاكها للطاقة في حالات الخمول و التي هي الحالة ذات الزمن الأطول في نمط استخدام الهواتف الذكية.

يهدف البحث إلى تطوير تقنية تستفيد من ميزات التقنيات الموجودة و تدمج فيما بينها مع مراعاة عدم تعارض هذه التقنيات مع بعضها.

طرائق البحث و مواده

تم التطرق في هذا البحث إلى مفهوم تعدد النوى و الأبحاث المتعلقة في هذا المجال و اخترنا مجال استهلاك الطاقة. و بعدها تمت دراسة أغلب تقنيات ادارة الطاقة في المعالجات متعددة النوى و شرح آلية عملها و حدودها و الأبحاث التي اجريت عليها لمعالجة قصورها (الفقدان في الأداء ، توفير الطاقة، الحالات الملائمة لكل تقنية) و المقارنة بينهم. و في النهاية تم اقتراح تقنية تستفيد من التقنيات مجتمعة و تتجنب مساوئها .

1 - لمحة عامة عن المعالجات

تزايد أداء المعالجات الصغرية بشكل أسي سنويا لذا ابتكرت تقنيات لتحقيق التوازي بدءا بالمعالجة بالتجزئة (pipeline) مرورا ببنية المعالج فائق التدرج (superscalar) وصولا إلى رقائق متعددة المعالجات او المعالجات متعددة النوى. تستفيد هذه البنية من وجود تعليمات او مهام مستقلة لا تعتمد معاملاتها على نتائج تنفيذ عليمات اخرى و تنفذها على التوازي. كما يمكنها استغلال وجود تفرع في سير عمل البرنامج. كانت هناك محاولات عديدة عبر السنين الماضية لتحسين التوازي في بنى المعالجات موضحة في القائمة التالية:

1-2 معالج الدورة الوحيدة: Single-cycle processor

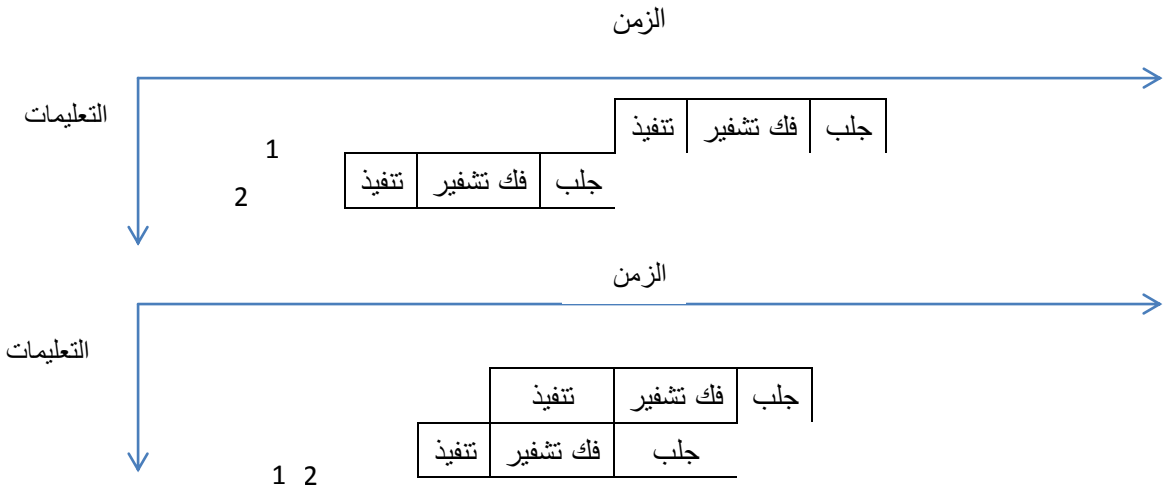
استعملت هذه التقنية في أوائل المعالجات و تعتمد على مبدأ أن تنفذ كامل التعليمة في نبضة ساعة واحدة. بمجرد أن تدخل تعليمة حيز التنفيذ على جميع التعليمات الاخرى الانتظار حتى تنتهي التعليمة من التنفيذ بشكل كامل و ينخفض أداء النظام بشكل عام بسبب وجود بعض التعليمات التي تستغرق وقتا طويلا في التنفيذ[3].

1-3 المعالجة بالتجزئة Pipeline:

بدلا من تنفيذ كامل التعليمة في نبضة ساعة واحدة يقسم التنفيذ الى مراحل. يعتمد هذا التقسيم على بنية المعالج. في كل مرحلة ينفذ جزء من تعليمة بشكل متوازي مع جزء آخر من تعليمة أخرى مثال : اذا كان لدينا pipeline مقسم الى ثلاث مراحل جلب التعليمة و فك التشفير و التنفيذ. يمكننا معالجة اجزاء من ثلاث تعليمات في

الزمن

نفس الوقت مثال: عند النبضة الثالثة حيث تكون التعليمة الأولى وصلت مرحلة التنفيذ بينما التعليمة الثانية في مرحلة فك التشفير و التعليمة الثالثة دخلت مرحلة الجلب. مما يقلل من زمن الانتظار بشكل ملحوظ كما في الشكل (1).



الشكل (1): الفرق بين معالج الدورة الوحيدة و معالج pipeline

و من ناحية اخرى يسبب ال pipeline بعض التأخير بسبب وجود مسجلات بين المراحل كما أن هناك مشاكل تتعلق بالاعتمادية بين البيانات.

1-4 المعالجة بالتجزئة العميقة Deep pipeline:

يعتمد على مبدأ زيادة عدد مراحل pipeline مؤديا بذلك إلى تزايد سرعة التنفيذ بسبب امكانية تنفيذ عدد اكبر من التعليمات في نفس الوقت. ال pipeline حاليا تمتلك 20 مرحلة و حدد هذا العدد بسبب الاعتمادية و بسبب الكلفة المنطقية التي تتمثل بزيادة عدد المسجلات التي تستعمل لتخزين القيم الوسيطة بين المراحل. و قد ظهرت عدة ابحاث للتخلص من مشاكل الاعتمادية مثل إعادة التوجيه و الإيقاف و إعادة تسمية المسجلات [6].

1-5 معالج فائق التدرج Super scalar:

من أهم مشاكل pipeline أنه على الرغم من قدرته على تنفيذ عدة تعليمات في وقت واحد و لكن في مراحل مختلفة من عمل التعليمة أي أنه لا يمكنه البدء بتنفيذ أكثر من تعليمة واحدة في كل نبضة. يتميز معالج super scalar بامتلاكه لعدة نسخ من مسار المعالجة (بما فيها وحدة الحساب و المنطق) مما يمكنه من البدء بتنفيذ عدد من التعليمات يساوي عدد نسخ هذا المسار. تم دمج التقنيتين لينتج معالج pipelined superscalar استعمل بكثرة في نهاية التسعينات. و من مساوئه شغله لمساحة كبيرة على الرقاقة و استهلاكه العالي للطاقة.

1-6 معالج خارج الصف: out of order

يفحص هذا المعالج مجموعة من التعليمات معا لاكتشاف المستقلة منها و تنفيذها على الفور. أي ان التعليمات لا تنفذ بنفس ترتيب ورودها. بمجرد أن تصبح معاملات التعليمة متوفرة تنفذ فورا دون الاكتراث لتسلسل تعليمات البرنامج. مما يمكن هذا المعالج من حل مشكلة الاعتمادية المرافقة لمعالجات pipeline و superscalar. تسوء هذا

التقنية مشكلة وحدة فحص جاهزية و اعتمادية التعليمات مما يضيف مساحة إضافية للرقاقة و استهلاك اضافي للطاقة.

1-7 معالج متعدد النوى:

يعمل المعالج متعدد النوى على مبدأ التوازي لتنفيذ سلاسل الاوامر بفعالية. مع الإشارة إلى أن العملية (Process) هي برنامج قيد التنفيذ حاليا و كل عملية تتكون من عدة سلاسل من الاوامر (Threads). تعطي الأنظمة التقليدية الاحساس بأنها تنفذ العمليات على التوازي و لكن الحقيقة أن هناك تبديل سريع يحدث بين سلاسل تعليمات العملية Threads (وهو أسرع بكثير من التبديل بين العمليات). البنية متعددة النوى هي الأكثر استفادة من قابلية سلاسل التعليمات للتنفيذ على التوازي، حيث تنفذ كل سلسلة على نواة مخصصة لها وهو ما يمثل التوازي الفعلي الذي يزيد من كفاءة النظام بشكل ملحوظ. تظهر في هذه البنية مشاكل مثل التصميم الملائم لهرمية الذاكرة و مدى صلاحية البيانات بين النوى و تصميم شبكة الارتباط الداخلية بين النوى و اعتمادية النوى على بعضها و إدارة الطاقة.

2 - تقنيات إدارة الطاقة:

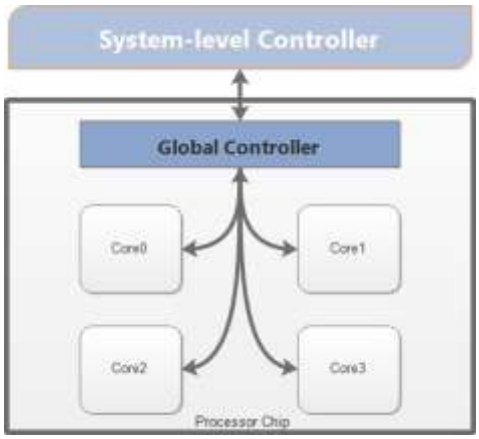
أصبحت إدارة الطاقة موضوعا أساسيا في تصميم معالجات متعددة النوى. حيث تكثر سلبيات الاستهلاك العالي للطاقة مثل الخواص الحرارية غير المستقرة للرقاقة. مؤثرة بذلك على أداء النظام مما يجعل تخفيض استهلاك الطاقة ذو اهمية بالغة. لا تحتاج سلاسل الاوامر التي تنفذ في نوى مختلفة كمية متساوية من الطاقة دائما فهناك مثلا اوقات انتظار بسبب الولوج للذاكرة مما يتطلب توفير في الطاقة المستهلكة اثناء الانتظار. لذلك كان من الضروري استعمال الضبط الديناميكي للطاقة المستخدمة في المعالجة لتحقيق التوازن بين الاداء و الطاقة اعتمادا على التحليل الفوري لحاجات الحمل.

تصنف التقنيات المتطورة لإدارة الطاقة تحت فئتين رئيسيتين، تفاعلية و تنبؤيه: في التقنيات التفاعلية تتفاعل التقنية مع تغييرات متطلبات الاداء في الحمل، أي أن الحمل قد يحتوي حالات تتطلب معالجة عالية الكثافة و حالات أخرى فيها انتظار عمليات (دخل/خرج) لا تحتاج طاقة، عندما تتغير حالة الحمل تستجيب التقنية وفقا لهذا التغير. تبرز في هذه التقنية مشكلة التأخير ما بين تغير حالة الحمل و تغيير مستوى أداء المعالجة مما قد يؤدي إما الى هدر طاقة او تأخر في الاستجابة لمتطلبات اداء عالي. من ناحية أخرى، تقنيات تنبؤ كالمذكورة في [7] تتخطى هذه المشكلة، تتنبأ هذه التقنيات بتغير حالة الحمل قبل حدوثه، بذلك تتخذ قرار التغيير في الوقت المناسب مما يؤدي إلى أداء أفضل و توفير أكبر في الطاقة. و لكن لا يوجد أحمال قابلة للتوقع بشكل مطلق لذا تستعمل التقنيات التفاعلية جنبا الى جنب مع تقنيات التنبؤ. و بما أن استعمال تقنيات التفاعل محتوم تم التركيز عليها في هذا البحث. تمت مناقشة بعض التقنيات الديناميكية لتحقيق أفضل إدارة للطاقة في المعالج متعدد النوى كما تمت مناقشة بعض مشاكل هذه التقنيات و كيف سعت الأبحاث السابقة الى التعامل مع هذه المشاكل.

تفترض جميع النظريات وجود متحكم طاقة في الشريحة مجهز بالدارات اللازمة لأداء عمله مدعوما بالبرمجيات اللازمة لتوجيهه نحو تطبيق تقنية أو خوارزمية. يظهر الشكل(2) عملية إدارة الطاقة من منظور عالي المستوى، متحكم الطاقة الخاص بالنظام يوجه متحكم الطاقة الخاص بالشريحة نحو قيمة طاقة معينة، يراقب متحكم الطاقة احصائيات الاداء للنوى و يتخذ قراره اعتمادا عليها. هذا القرار يختلف باختلاف الخوارزمية المتبعة (مثلا تقنية DVFS تعدل مستويات الجهد و التردد، بينما تقنية power gating تقطع الطاقة تماما عن الاجزاء الخاملة).

تقيَّم التقنيات وفق مقاييس متنوعة، و يعد مقياس طاقة / التعليمة (EPI: Energy per instruction) المقترح بواسطة [8] من المقاييس الهامة، فالتقنية التي تتمكن من تخفيض EPI تعد أكثر كفاءة. الهدف الأساسي لجميع التقنيات هو زيادة التعليمات في الدورة (IPC: instruction per cycle) مع الحفاظ على EPI أصغر ما يمكن و تحقيق هذا التوازن هو الأهم في جميع ابحاث ادارة الطاقة.

يمكن اعتبار نظام التحكم بإدارة الطاقة حلقة مغلقة ذات تغذية عكسية. مدخل النظام هو كمية الطاقة المحددة من قبل متحكم النظام (الخارجي)، يضبط متحكم الشريحة (الداخلي) بعض البارامترات مثل الجهد و التردد اعتمادا على مراقبة العمليات (التغذية العكسية) من قبل كل نواة بشكل منفرد. إن مراقبة استهلاك الطاقة بشكل مستمر هو موضوع هام و حساس حيث يجب على أية تقنية توفير طاقة ان تراقب الطاقة المستهلكة لتوجيه قراراتها.



الشكل (2): منظور مبسط لعملية إدارة الطاقة

2-1 تقنية التدرج الديناميكي للجهد و التردد: Dynamic voltage and frequency scaling(DVFS)

2-1-1 المبدأ الأساسي:

يمكن التحكم بكمية الطاقة المستهلكة بتدريج قيمتي التردد والجهد اعتمادا على قانون استهلاك الطاقة(1):

$$P=CV^2F \qquad (1)$$

حيث P هي الطاقة و C سعة التبديل و V الجهد و F تردد العمل. و قد تم تقديمها تجاريا تحت مسميات متعددة فقد سمتها شركة انتل: SpeedStep، و سمتها شركة AMD: تقنية PowerNow. الفكرة الأساسية هي تدريج قيم الجهد و التردد إلى قيم محددة مسبقا لتحقيق الأداء و الاستهلاك المناسبين للحمل الحالي. تختلف آلية التطبيق باختلاف نمط التزامن بين نوى المعالج، في البنية المتزامنة يكون التردد و الجهد واحدا لكل النوى و تتغير قيمته حسب الحمل. في البنية اللامتزامنة تعمل كل نواة بتردد و جهد مختلفين حيث تعاير النواة ترددها و جهدها وفقا للحمل المطبق عليها. في حالة الحمل المتوازي يعمل أكبر عدد من النوى بتردد منخفض بينما في حالة حمل تسلسلي يجب تشغيل عدد أقل من النوى بتردد عالي.

المتحكم على مستوى النظام (الخارجي) يوجه المتحكم على مستوى الرقاقة (الداخلي) بقيم الجهد المرغوبة. يراقب المتحكم الداخلي تردد و جهد كل نواة. اعتمادا على هذه المدخلات يغير المتحكم الداخلي الجهد و التردد إلى المستوى المطلوب. يطبق ذات المبدأ في جميع تقنيات ادارة الطاقة و يقع الاختلاف في الخوارزمية نفسها.

2-1-2 حساب التردد و الجهد المناسبين:

تختار بعض الأنظمة قيم جهد و تردد محددة مسبقا و لا تعتبر هذه الطريقة فعالة حيث قد لا تكون القيمة المطلوبة تطابق تماما احدى القيم المحددة مسبقا. قدم Kamga في [9] اقتراحا حول الحساب الدقيق للتردد المطلوب للحمل الحالي اعتمادا على العتبة العليا و الدنيا و عدد مرات حدوث كل منهما. خلصت الطريقة إلى أن التردد الأنسب هو:

$$f_{host} = (f_{high} * t_{high}) + (f_{low} * t_{low}) / t_{high} + t_{low} \qquad (2)$$

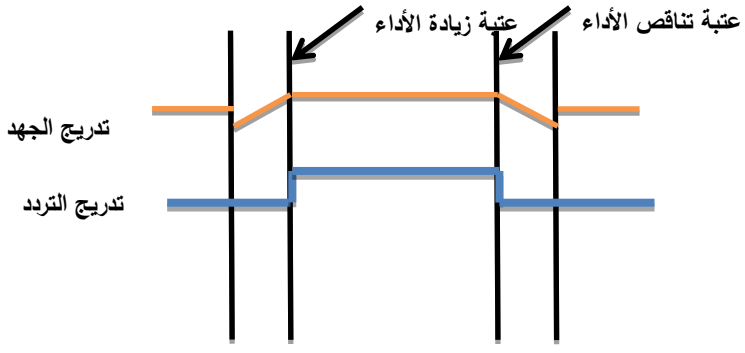
حيث f_{host} هو التردد المطلوب و f_{high} هو عتبة التردد العليا و t_{high} عدد مرات ظهورها و كذلك الأمر بالنسبة للعتبة الدنيا.

2-1-3 مستويات تطبيق DVFS:

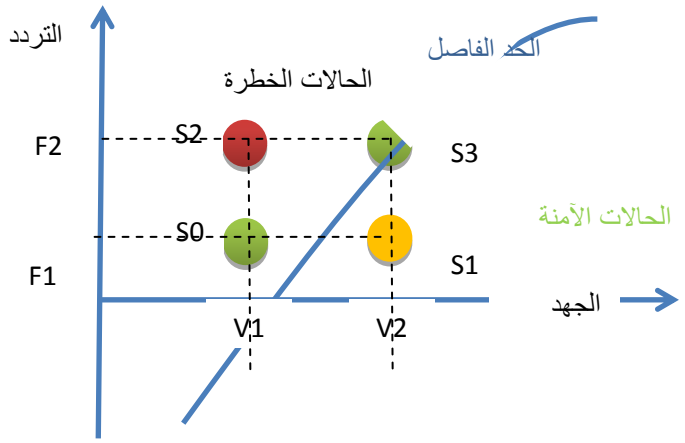
يمكن تطبيق DVFS على مستوى الرقاقة او على مستوى النواة و تطبيقها على النواة أكثر مرونة حيث يمكن لكل نواة ان تعمل بتردد و جهد مختلف و لكن بالمقابل سيزداد عدد منظمات الجهد في الرقاقة. و أما تطبيق DFVS على كامل الرقاقة سيقلل عدد منظمات الجهد و لكنه سيحد من المرونة حيث سيطبق الجهد نفسه على كل النوى متجاهلا الاحتياجات الخاصة لكل نواة. إن حساب الجهد و التردد المناسب لكل النوى معا أمر بالغ الصعوبة و التعقيد. قدم Kolpe في المرجع [10] تقنية وسيطة تدعى عنقدة DVFS و التي تجمع النوى في مجالات DVFS مختلفة و تطبقها على أساس العنقود و يمكن اختصار الخوارزمية بثلاث خطوات: إيجاد الجهد و التردد المثالي لكل نواة – ايجاد النوى المتشابهة و وضعها في عنقود – تقييم الحل بإيجاد التردد و الجهد المثالي لكل عنقود و مقارنته بالإعدادات الحقيقة للعنقود. أثبتت الطريقة تحسنا في النتائج و لكنها أقل فاعلية عند عدد كبير من العناقيد.

2-1-4 زمن تعديل الجهد و التردد:

يتطلب تعديل الجهد و التردد بعض التأخير للوصول للمستوى المطلوب. و تظهر المشكلة في أن تعديل التردد أسرع من تعديل الجهد، يوضح الشكل (3) التأخير اللازم لتعديل الجهد حتى يستطيع زيادة أو تخفيض الأداء مما قد يضع المعالج في حالة خطرة لا يستطيع فيها الجهد المطبق دعم التردد المطلوب مؤديا بذلك إلى إيقاف المعالج بشكل كلي. يظهر الشكل (4) العلاقة بين الجهد و التردد خلال DVFS حيث يمكن تقسيم فضاء الحالة إلى ثلاث عينات: S2 كل الحالات فوق الحد الفاصل هي حالات خطرة لا يستطيع فيها الجهد دعم التردد المطلوب. S1 الحالات تحت الحد الفاصل هي حالات مستهلكة للطاقة دون فائدة عملية. (S0,S3) الحالات التي تنتمي لجوار الحد الفاصل هي حالات آمنة للمعالج موفر للطاقة[11].



الشكل (3): مقارنة سرعة تبديل الجهد بسرعة تبديل التردد



الشكل (4): العلاقة بين الجهد و التردد خلال التعديل الديناميكي

و لتجاوز هذه المشكلة يمكن البدء بتدريج الجهد و إيقاف التطبيق عن العمل حتى انتهاء تغيير الجهد ثم البدء بتغيير التردد. و يبدو التأخير واضحا في هذه الطريقة. قدمت عدة أبحاث لتقليل تأخير DVFS حيث ينصح Lai في [11] بتجنب كثرة الانتقال بين الحالات و بتدريج التردد بمفعول رجعي بمعنى عدم إيقاف التطبيق عن العمل أثناء تعديل الجهد و إنما يبقى التطبيق يعمل بالتردد السابق حتى انتهاء عملية تعديل الجهد. برغم أن هذه الطريقة تقلل من التأخير إلا انها ستؤدي إلى المرور في حالة S1 المستهلكة للطاقة أثناء تعديل الجهد. طبعا هذه الطريقة فعالة في حال الارتفاع في قيم الجهد و التردد و لكنها لا تناسب الانخفاض حيث أن بقاء التطبيق يعمل بالتردد السابق بينما ينخفض الجهد سيؤدي بالمعالج إلى المنطقة الخطرة.

2-1-5 حدود تقنية DVFS:

مستوى التطبيق : تظهر بعض المشاكل اعتمادا على المستوى الذي تطبق عليه DVFS. فإذا كان التطبيق على مستوى النواة تزداد التكلفة و تقل المرونة عند تزايد أعداد النوى. و عند تطبيقها على مستوى المعالج كاملا فإنها تتجاهل الحاجات الفردية للنوى. و على مستوى العنقود تقل الكفاءة عند تزايد عدد العناقيد إلى حد معين.

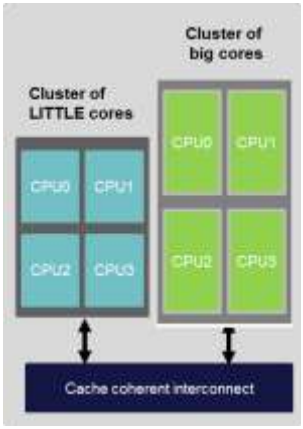
تأخير الانتقال بين الحالات: إن بطء تغيير الجهد يجعل الانتقال بين الحالات بطيئا و خطرا على المعالج في بعض الأحيان. يسبب هذا التأخير مشكلة كبيرة خصوصا عندما تتغير الحاجة إلى هذا الانتقال قبل الوصول إليه مما يجعله دون فائدة.

2-2 تقنية النوى غير المتناظرة: Asymmetric cores

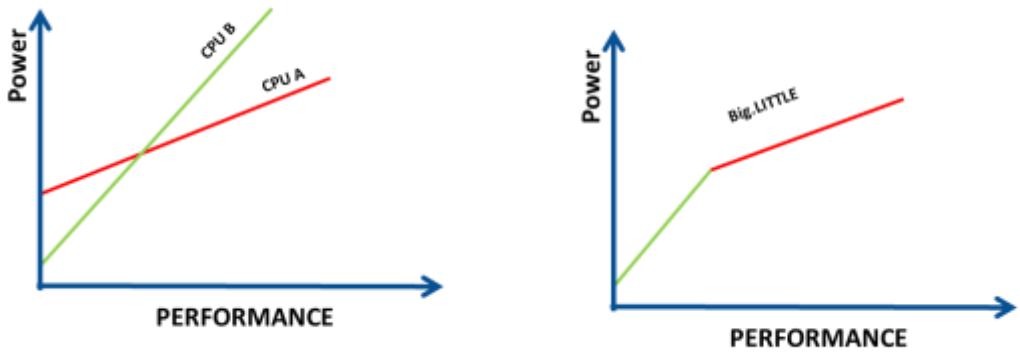
هي تصميم معالج متعدد النوى متغايرة الخواص بمجموعة تعليمات قابلة للتنفيذ على جميع النوى بشكل متطابق حيث يختلف أداء النوى و استهلاكها عن بعض، وهو مبدأ هام جدا في مجال توفير الطاقة. هناك نوعان من النوى، كبيرة تكون فائقة التدرج عميقة pipeline تنفذ خارج الترتيب تدعم الأحمال التسلسلية العالية الكثافة– نوى صغيرة بسيطة تنفذ التعليمات بالترتيب مزودة ب pipeline قصير مناسبة للأعمال المتوازية. و من أشهر تطبيقات النوى غير المتناظرة تقنية big.LITTLE لشركة ARM و تقنية 1+4 لشركة NVidia

2-2-1 تقنية big.LITTLE:

اطلقتها شركة ARM سنة 2011 حيث دمجت فيها معالج رباعي النوى كبير cortex A15 مع معالج رباعي النوى صغير Cortex A7 الشكل (5). حيث أن كلا المعالجين صمما بنفس المعمارية v7A. أي أن المعالجات متطابقة بنيويا و بطريقة تنفيذها للتعليمات و يمكن ترحيل التعليمات بين المعالجين دون الحاجة لإعادة ترجمة التعليمة رغم وجود اختلاف في الاداء يعمل فيه المعالج الكبير على معالجة الاحمال التسلسلية الكثيفة المستهلكة للطاقة بينما يخصص المعالج الصغير للأعمال البسيطة المتوازية. تبنت شركة سامسونغ هذه البنية في معالجها ثماني النوى Exynos 5[12]. على اعتبار أن استهلاك الطاقة الخاص بتنفيذ التعليمات متعلق جزئيا بعدد مراحل pipeline الواجب قطعها فالفرق الأعظم بين استهلاك المعالجين للطاقة هو في عدد مراحل pipeline. بالإضافة إلى اختلاف بنيوي على مستوى أعمق وهو الترانزستورات. الترانزستورات التي تمتاز بسرعة تبديل كبيرة عند مستوى متوسط من الجهد تستهلك تيار تسريب عالي في حالة الخمول كما المعالج A في الشكل (6). أما النوع الآخر من الترانزستورات في المعالج B التي تمتاز بتيار تسريب ضئيل فإنها تحتاج إلى وقت أطول في عملية التبديل و لزيادة هذه السرعة تستهلك طاقة كبيرة بالإضافة إلى مشاكل السخونة. تم حل هذه المشاكل بجعل المعالجين على رقاقة واحدة حيث يعمل المعالج الصغير حتى عتبة اداء معين ثم ينتقل التنفيذ إلى المعالج الكبير.

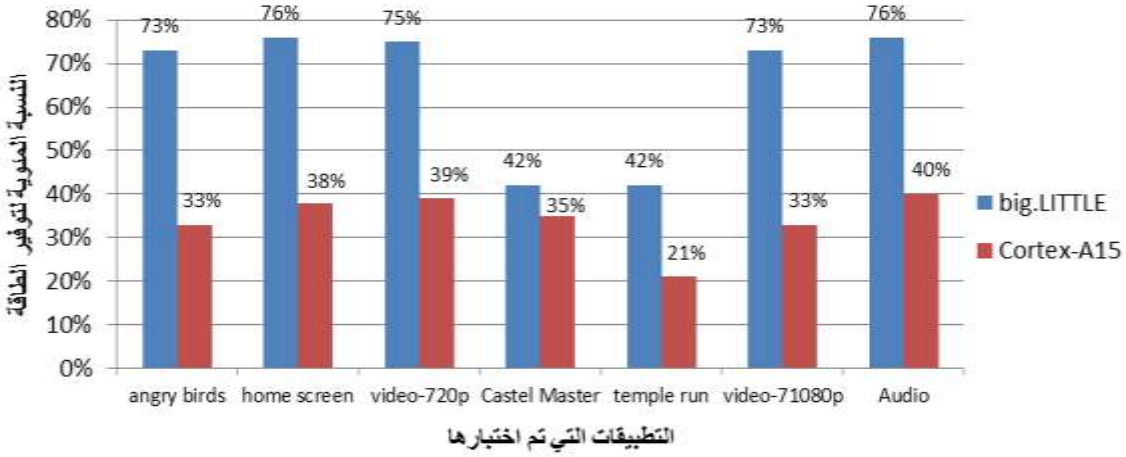


الشكل (5) : بنية معالج big.LITTLE



الشكل (6) : مبدأ عمل big.LITTLE

يظهر الشكل(7) الفرق في توفير الطاقة بين استخدام معالج big.LITTLE و بين استخدام معالج Cortex-A15 منفردا من خلال حالات استعمال متنوعة.



الشكل (7): مقارنة توفير الطاقة لمعالج A15 مع معالج بتقنية big.LITTLE

كما أن لدى شركة NVidia نموذجها الخاص عن المعالجات متعددة النوى غير المتناظرة يحوي خمس نوى، أربع منها عالية الاداء مستهلكة للطاقة بالإضافة لنواة واحد تسمى المرافقة موفرة للطاقة للقيام بالأعمال التي لا تتطلب قدرا كبيرا من المعالجة في حالة الخمول.

تستعمل النوى غير المتناظرة على نطاق واسع في الأجهزة المحمولة لتقليل استهلاكها. يمكن تطبيق DVFS على هذه البنى من المعالجات. و من المتوقع رؤية المزيد من هذه التشكيلات من النوى المتغايرة الخواص حيث أثبتت فعالية في الاداء و توفير الطاقة في جميع الأجهزة المحمولة و الأنظمة المستقلة.

2-2-2 المقاطع الحرجة و جدولة سلاسل التعليمات:

أهم مشاكل النوى غير المتناظرة تكمن في الجدولة أي اسناد سلسلة التعليمات Thread معينة إلى النواة الأنسب من ناحية الاداء و الطاقة معا. و إن اختلاف بنية المعالجات يسبب اختلاف في طرق الجدولة بسبب اختلاف المصادر و طريقة استعمالها و اختلاف مجالات الطاقة بينها. الاختيار الخاطئ لتقنية الجدولة سيسبب تأثيرا عكسيا

على الأداء. قدمت الدراسة [13] مخطط جدولة يقترح اسناد المهمة ذات زمن التنفيذ الأعلى إلى النواة الأسرع بينما اقترح Becchi في [14] استعمال IPC كمقياس لإسناد السلاسل للنوى.

استعملت بعض الأبحاث تقنية تحديد المقاطع الحرجة في سلسلة التعليمات اعتماد على آلية تسجيل نقاط. يتخذ قرار التحسين اعتمادا على نقاط كل سلسلة تعليمات. و تدعي الدراسة انه بتطبيق هذا المقياس لتعديل تردد النوى غير المتناظرة يوفر 28.13% في استهلاك الطاقة مع حد أعلى لانخفاض الأداء مقدرا ب 7.1%. تعتمد النظرية المقدمة في هذا البحث على مقياس IPC تعليمة في الدورة لكل سلسلة تعليمات في أوقات مختلفة. حيث تفترض أن زمن التنفيذ مقسما إلى أجزاء متساوية ولكل سلسلة تعليمات قيمة IPC في كل قسم. و تحسب نقاط السلسلة (Thread) في قسم معين بتقسيم IPC السلسلة على IPC جميع السلاسل في نفس القسم من الزمن. و تحسب النقاط النهائية لكل سلسلة بجمع كل النقاط في الأقسام الزمنية. يمكن صياغة القانون على الشكل التالي:

$$\sum_t \left(\frac{IPC_i^t}{\sum_i IPC_i^t}\right) \qquad (3)$$

حيث i رقم التعليمة و t رقم المقطع الزمني. السلسلة ذات النقاط الأعلى هي الأكثر حرجا بين السلاسل الأخرى و بذلك فإن تسريع تنفيذها سيقلل من زمن التنفيذ العام[15].

2-2-3 النوى غير المتناظرة بأكثر من نوعين:

بالإضافة إلى استعمال نوعين من النوى في المعالج متعدد النوى. تقترح الدراسة [16] أن تحتوي الرقاقة على عدة انواع من النوى و ليس فقط نوعين. تقرر هذه الدراسة أي النوى أمثل لتنفيذ تطبيق و ترحله إليها. يقترح في المثال معالج خماسي النوى مختلفة البنية و يدعي أن النتائج تشير إلى تحسن في استهلاك الطاقة بمقدار ثلاث أضعاف مقابل خسارة 18 % من الأداء.

2-2-4 حدود تقنية النوى غير المتناظرة

نستنتج من الفقرات السابقة أن لتقنية متعددة النوى غير المتناظرة بعض الحدود. أهمها أن عدد النوى الكبيرة و الصغيرة ثابت في مرحلة التصميم لا يمكن تعديله مما يقلل من ملائمة البنية للتطبيقات المختلفة.

المحدودية الهامة الأخرى هي التأخير الناتج عن هجرة المهام بين النوى و بسبب هذه الهجرة تظهر مشكلة اخرى هي محلية البيانات بسبب أن لكل نواة مستوى أو اكثر من الذاكرة المؤقتة الخاصة به. فعند تغيير المهمة للنواة التي تنفذ عليها يجب ان تهاجر معها بياناتها من ذاكرة مؤقتة لأخرى أو تجلب مرة ثانية من الذاكرة الرئيسية مما يسبب تأخير اضافيا يحد من تحسن الاداء.

2-3 تقنية حركة سلاسل التعليمات: Thread motion

المبدأ الأساسي: تعتبر هذه التقنية تحسينا ل DVFS. حيث تفترض أن مستوين فقط للجهد و التردد كافيين لتحسين الأداء. و تقوم فكرة حركة سلاسل التعليمات على امتلاك المعالج لنوى صغيرة تعمل بمستوين مختلفين من التردد و الجهد. عند تنفيذ التطبيقات، تقرر الخوارزمية أي نواة تعمل بالتردد و الجهد المناسب للمهمة و ترحله اليها بدلا من تغيير جهد و تردد النواة نفسها. هذه التقنية تمكن التطبيقات من الهجرة إلى نواة ذات تردد و جهد أعلى أو أقل حسب الحمل. أي إذا امكن التطبيق الاستفادة من جهد و تردد أعلى تعمل بهما نواة أخرى تقوم الخوارزمية بتبديل محتوى النواتين[17].

حدود التقنية : تتلخص بكونها موجهة إلى بنية نوى بسيطة متغايرة و تظهر النتائج تحسنا في الاداء 20% عوضا عن استخدام DVFS العادي مقابل الاستهلاك ذاته.

2-4 تقنية النوى متغيرة الحجم Variable size cores

المبدأ الأساسي: أساس النظرية هو تصميم نواة كبيرة معقدة يمكن اختصارها فيما بعد لتصبح نواة صغيرة. يمكن تحقيق هذا من خلال الإلغاء الديناميكي لبعض وحدات التنفيذ و دمج مراحل pipeline. هذه التقنية تعتمد على خوارزمية power gating المعروفة وهي تعطيل المصادر التي ثبت خمولها لعدد محدد من نبضات الساعة. في حال وجود حمل تسلسلي سيفيد وجود نوى كبيرة معقدة تسرع من الانجاز. و في حالة الحمل المتوازي فإن تحويل هذه النوى إلى نوى بسيطة باختصار بعض اجزائها سيكون ذو مردود جيد على الاداء و استهلاك الطاقة[18].

حدود التقنية: يبدو واضحا أن للتقنية حدود كثيرة منها :

خطأ التنبؤ : كما شرح سابقا فإن التقنية تعتمد على إلغاء المصادر التي ثبت خمولها لعدد من نبضات الساعة و بسبب هذا المبدأ بالتنبؤ يمكن أن تغلق مصدرا مباشرة قبل أن تعود الحاجة إليه مرة أخرى مما يؤثر سلبا على الأداء و استهلاك الطاقة .

قلة توفير الاستهلاك: صحيح أن إغلاق بعض المصادر في حالة عدم الحاجة إليها يوفر طاقة بالتأكيد إلا أن هذه الطاقة ليست بذات تأثير كبير و هي قليلة مقارنة بالطاقة الموفرة من خلال استعمال تقنية DVFS .

2-5 دمج النوى: Core fusion

المبدأ الأساسي: يصمم المعالج على شكل نوى صغيرة تندمج لتشكل نوى أكبر لدعم الاحمال التسلسلية عند الحاجة . ولا تتطلب التقنية أي جهد برمجي إضافي ولا مترجم مصمم خصيصا. يتناسب دمج النوى مع تنوع الأحمال. حيث توزع الأحمال المتوازية على النوى الصغيرة و تدمج النوى الصغيرة ديناميكيا لتشكل نوى كبيرة تتناسب مع الأحمال التسلسلية. التفاصيل الكاملة حول هذه البنية موجودة في [4].

حدود التقنية: هي أن النوى الكبيرة المدمجة تستهلك طاقة أكبر من النوى الكبيرة التقليدية لأن هناك تأخر إضافيا بين مراحل pipeline النواة المدمجة. كما أن دمج النوى الصغيرة يشكل تأخير لنقل البيانات من الذاكرة المؤقتة الخاصة بها أو مسحها و استدعاء البيانات من جديد للذاكرة المؤقتة للنواة الكبيرة المدمجة.

النتائج و المناقشة:

اعتمادا على ما سبق و وفقا للجدول (1) مقارنة بين تقنيات إدارة الطاقة في المعالجات المتعددة النوى:

الجدول (1): مقارنة بين تقنيات إدارة الطاقة في المعالجات المتعددة النوى

	DVFS	نوى غير متناظرة	حركة سلاسل التعليمات	نوى متغيرة الحجم	دمج النوى
الفكرة	تدريج قيم الجهد و التردد وفقا للأداء المطلوب	تنفيذ الاحمال المتوازية على نوى صغيرة و استخدم الكبيرة لتنفيذ الأحمال التسلسلية	نقل سلاسل التعليمات بين النوى التي تعمل بمجالين مختلفين فقط	تشغيل و إيقاف المصادر حسب الحاجة إليها	دمج النوى الصغيرة المستعملة لتنفيذ مهام متوازية للحصول على نوى كبيرة و تنفيذ المهام التسلسلية

الميزات	فعالية في توفير الطاقة مع انخفاض صغير في الاداء. سهلة التطبيق.	نتائج فعالة في توفير الطاقة. تجاوب مع تنوع التطبيقات.	سرعة التغيير. نتائج فعالة في توفير الطاقة (اعتمادا على DFVS).	بسيطة التطبيق يمكن استخدمها مع تقنيات توفير طاقة أخرى.	ملائمة ممتازة لتنوع التطبيقات. ليست بحاجة إلى جهد برمجي إضافي.
القصور	التأخير الناتج عن الانتقال بين حالات الطاقة. عمق مستوى التفاصيل التي تطبق عليها التقنية يؤثر على الكلفة و الأداء.	عدد النوى الكبيرة و الصغيرة ثابت من مرحلة التصميم. التأخير الناجم عن ترحيل المهام بين النوى. اختيار تقنية الجدولة الأنسب.	استخدامها محصور على بنية متعددة النوى المتغايرة الخواص.	توفير بسيط للطاقة عند استخدامها وحيدة. خطأ التنبؤ بإغلاق مصدر سيهدر طاقة.	التأخير في النوى المدمجة أكبر من التأخير في النوى التقليدية و التأخير الناتج عن انتقال البيانات بين النوى الصغيرة و الكبيرة عند التبديل بينهما
نسبة التحسن	لا يمكن قياس النسبة فهي تقنية عامة تختلف باختلاف البنية و مستوى التطبيق	أعلى نسبة توفير، 70% في حالة استخدام يومي عادي لجهاز محمول	20 % توفير في الاستهلاك عند استخدام خوارزمية تبديل ملائمة مع تغيير الحمل [16].	16% توفير في الاستهلاك ما لم يتجاوز خطأ التنبؤ 30% [17].	تحسن IPC تصل إلى 73% في أحسن حالاته [4].

إن التقنية الهجينة المقترحة تحقق التوازن الأفضل بين قوة الأداء و تخفيض استهلاك الطاقة. حيث تقوم هذه التقنية على استعمال DVFS العنقودية مع تدرج التردد العكسي في معالج متعدد النوى غير المتناظرة متعددة الأنواع و استخدام تقنية النقاط في عملية جدولة سلاسل التعليمات. كما يمكن استخدام تقنية power gating في حالة الاستخدام القليل للمعالج. يتحكم بالتقنية من قبل متحكم ذو طبيعة متأقلمة وفقا لبارامترات مثل (نوع الحمل ، الاستخدام الحالي للنوى ، امكانية تحقيق التوازي، الأداء الحالي) تمكننا من تنفيذ هذه التقنية بكفاءة.

مثال: عندما يكون الحمل عالي التوازي يتم تثبيت تردد النوى الصغيرة و يقلل تردد النوى الكبيرة و تستخدم جميعها لتنفيذ البرنامج. أما في حالة الحمل التسلسلي يرفع تردد النوى الكبيرة. التقنية المقترحة قابلة للتطوير المستقبلي و اثبات نتائجها بواسطة المحاكاة .

الاستنتاجات و التوصيات

يتضح من المناقشة أنه لا يوجد طريقة مثالية نهائية لإدارة الطاقة في المعالج. و يعتمد اختيار تقنية إدارة الطاقة على وجود امكانية لتغيير البنية كما يعتمد اختيار التقنية على مستوى الاداء المقبول خسارته و كمية الحمل المتوقع انجازه في هذا المعالج.

أفضل الحلول لتحسين فعالية تقنيات إدارة الطاقة في المعالج هو دمج بعض التقنيات معاً، على سبيل المثال نرى أن تطبيق تقنية النقاط في الجدولة واعدة جدا. كذلك عند تطوير تقنيات جديدة من يجب الأخذ بعين الاعتبار توافقها قدر الامكان لا تتعارض مع تقنيات اخرى .

المراجع:

[1]- World Economic Forum. The Global Information Technology Report 2015

[2]- Moore Gordon E. *Cramming more components onto integrated circuits*, vol. 12; 1965.

[3]- Kunle Olukotun, Basem A. Nayfeh, Lance Hammond, Ken Wilson, and Kunyung Chang. *The case for a single-chip multiprocessor*. ACM Sigplan Notices 1996.

[4]- Amdahl Gene M. *Validity of the single processor approach to achieving large scale computing capabilities*. In: Proceedings of the April 18–20, 1967, spring joint computer conference. ACM; 1967.

[5]- Ipek Engin, Meyrem Kırman, Nevin Kırman, and Jos´e F. Martınez. *Core fusion: accommodating software diversity in chip multiprocessors*. ACM SIGARCH computer architecture news, vol. 35(2). ACM; 2007.

[6]- Karthik Natarajan, Heather Hanson, Stephen W. Keckler, Charles R. Moore, Doug Burger. *Microprocessor pipeline energy analysis*. In International Symposium on Low Power Electronics and Design, 2003. ISLPED '03.; 25-27 Aug. 2003: 282-287.

[7]- William Lloyd Bircher, Lizy John. *Predictive Power Management for Multi-Core Processors*. WEED 2010 - Workshop on Energy-Efficient Design, Jun 2010, Saint Malo, France.

[8]- Horowitz Mark, Indermaur Thomas, Gonzalez Ricardo. *Low power digital design*. In: Low power electronics, 1994. Digest of technical papers. IEEE; 1994.

[9]- Mayap Kamga, Christine Larissa and Hagimont, Daniel. *Virtual processor frequency emulation*. (2013) In: 19th International Conference on Parallel and Distributed Processing Techniques and Applications. (Las Vegas ،United States); July 2013.

[10]- Tejaswini Kolpe. *Power Management in Multicore Processors through Clustered DVFS*. UNIVERSITY OF MINNESOTA. July, 2010.

[11]- Zhiquan Lai, King Tin Lam, Cho-Li Wang, Jinshu Su, Youliang Yan and Wangbin Zhu. *Latency-aware dynamic voltage and frequency scaling on many-core architectures for data-intensive applications*. In: 2013 international conference on cloud computing and big data (CloudCom-Asia). IEEE; 16-19 Dec. 2013: 78 - 83.

[12]- ARM (2013) big.LITTLE Technology: The Future of Mobile. White paper. Available at Internet: http://www.arm.com/files/pdf/big_LITTLE_Technology_the_Futue_of_Mobile.pdf [Retrieved 2015-05-25]

[13]- Lakshminarayana Nagesh B et al. *Age based scheduling for asymmetric multiprocessors*. In: Proceedings of the conference on high performance computing networking, storage and analysis; 14-20 Nov. 2009: 1 - 12.

[14]- Becchi Michela, Crowley Patrick. Dynamic thread assignment on heterogeneous multiprocessor architectures. In: Journal of Instruction-Level Parallelism 10 (2008): 1-26.

[15]- Manakkadu Sheheeda, Dutta Sourav, Botros Nazeih M. Power aware parallel computing on asymmetric multiprocessor. In: System-on-Chip Conference (SOCC), 2014 27th IEEE International;2014:35-40

[16]- V. Petrucci , O. Loques ; D. Mossé ; R. Melhem ; N. Gazala ; S. Gobriel. Thread Assignment Optimization with Real-Time Performance and Memory Bandwidth Guarantees for Energy-Efficient Heterogeneous Multi-core Systems. Real-Time and Embedded Technology and Applications Symposium (RTAS), 2012 IEEE 18th: 263 - 272.

[17]- Rangan Krishna K, Wei Gu-Yeon, Brooks David. Thread motion fine-grained power management for multi-core systems. In the 36th annual international symposium on Computer architecture; 2009:302-313.

[18]- Efthymiou Aristides, Garside Jim D. Adaptive pipeline depth control for processor power-management. IEEE International Conference;2002: 454 – 457